

4231/ITS/14/91 ✓

# PERENCANAAN DAN PEMBUATAN ALAT KONTROL KERJA MESIN BUBUT DENGAN MENGGUNAKAN MIKROKOMPUTER M8035L



Oleh :

**ISRAMSYAH**  
**NRP. 2852200383**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
1990**

P8E  
621.398 1  
Isr  
P-1  
1990

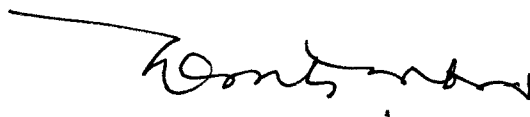
H.  
02/TA

# **PERENCANAAN DAN PEMBUATAN ALAT KONTROL KERJA MESIN BUBUT DENGAN MENGGUNAKAN MIKROKOMPUTER M8035L**

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar  
Sarjana Teknik Elektro  
Pada  
Bidang Studi Teknik Elektronika  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Mengetahui / Menyetujui  
Dosen Pembimbing**



**Ir. NAWANTOWIBOWO**

**S U R A B A Y A**

**NOPEMBER, 1990**

## ABSTRAK

---

Dalam bidang industri, mesin bubut digunakan untuk menghasilkan benda-benda kerja yang berbentuk silinder.

Dalam tugas akhir ini akan dibuat pengontrolan posisi dari tool dan pengontrolan posisi dari feeding (pemakanan). Data model yang dibuat di IBM PC ditransfer ke-RAM dan selanjutnya data tersebut diproses oleh kontroller type 8035. Data di-RAM diambil oleh kontroller dan dibandingkan dengan posisi tool, jika tidak sama kontroller akan menggerakkan motor yang mengatur posisi tool sampai posisi tool sama dengan data yang disimpan di-RAM. Untuk pembuatan data model dan transfer data model ke-RAM digunakan bahasa tingkat tinggi yaitu Pascal. Sedang untuk kontroller digunakan bahasa tingkat rendah yaitu bahasa assembly dari kontroller 8035.

Hasil tugas akhir ini dapat digunakan untuk mengontrol posisi dari tool sehingga akan mempermudah pekerjaan dari operator, yang biasanya proses pembubutan memerlukan ketelatenan dan ketelitian yang sangat tinggi.

## KATA PENGANTAR

---

Puji syukur kami panjatkan Kehadirat Tuhan Yang Maha Esa telah melimpahkan berkat dan rahmatNya, sehingga penulis dapat menyelesaikan tugas akhir yang berjudul :

### **"PERENCANAAN DAN PEMBUATAN ALAT KONTROL KERJA MESIN BUBUT DENGAN MENGGUNAKAN MIKROKOMPUTER M8035L"**

Adapun tugas akhir ini mempunyai beban kredit sebanyak 6 SKS (Satuan Kredit Semester) dan merupakan pelengkap persyaratan guna memperoleh gelar sarjana teknik elektro pada fakultas teknologi industri jurusan teknik elektro Institut Teknologi Sepuluh Nopember Surabaya.

Dalam penyusunan tugas akhir ini, penulis melakukan penyusunan berdasarkan pada teori-teori yang telah didapat selama kuliah, literatur, bimbingan dari dosen pembimbing, dosen wali serta pihak-pihak lain yang telah membantu dan mendorong semangat penyusunan.

Penulis berharap agar Tugas Akhir ini dapat memberikan banyak manfaat bagi para pembaca pada umumnya serta mahasiswa elektro pada khususnya.

Akhir kata tak lupa penulis menyampaikan banyak terima kasih yang sebesar-besarnya atas bimbingan, nasihat serta dorongan dari :

- Ir. Nawantowibowo sebagai dosen pembimbing dan dosen wali.
- Rekan-rekan mahasiswa bidang studi Elektronika yang telah banyak membantu.
- Serta semua pihak yang telah banyak membantu terwujudnya tugas akhir ini.

Surabaya, Nopember 1990

Penul s.

## DAFTAR ISI

BAB	HALAMAN
ABSTRAK .....	iv
KATA PENGANTAR .....	v
DAFTAR ISI .....	vi
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	x
I. PENDAHULUAN .....	1
I.1. LATAR BELAKANG .....	1
I.2. PERMASALAHAN .....	3
I.3. PEMBATAHAN MASALAH .....	3
I.4. METODOLOGI .....	4
I.5. LANGKAH-LANGKAH PEMBAHASAN .....	4
II. TEORI PENUNJANG .....	5
II.1. PENDAHULUAN .....	5
II.2. MESIN BUBUT .....	5
II.2.1. TOOL MESIN BUBUT .....	6
II.3. MIKROKOMPUTER SERPIH TUNGGAL 8035 ....	7
II.3.1. ARSITEKTUR .....	8
II.3.2. PROGRAM MEMORI .....	10
II.3.3. DATA MEMORI .....	11
II.3.4. INPUT / OUTPUT .....	13
II.3.5. PROGRAM COUNTER DAN STACK ...	16

11.3.6.	PROGRAM STATUS WORD .....	18
11.3.8.	INTERUPT .....	20
11.3.8.1	WAKTU INTERUPT .....	21
11.3.9.	PEWAKTU / PENCACAH .....	21
11.3.9.1	SEBAGAI PENCACAH .....	23
11.3.9.2	SEBAGAI PEWAKTU .....	24
11.3.10.	CLOCK DAN RANGKAIAN PEWAKTU..	24
11.3.11.	RESET .....	26
11.3.12.	PENJELASAN PIN-PIN .....	27
11.4.	MEMORI .....	30
11.4.1.	ROM (READ ONLY MEMORY) .....	30
11.4.2.	RAM (RANDOM ACCESS MEMORY) ..	31
11.4.3.	PENGAMBILAN DATA DARI MEMORI	32
11.4.4.	PENGIRIMAN DATA KE MEMORI ...	33
11.5.	INTERFACING KE IBM PC .....	34
11.5.1.	SLOT PERLUASAN IBM PC .....	35
11.5.2.	PENGALAMATAN I/O PORT PADA IBM PC .....	38
11.6.	OPTOCOUPLER .....	42
11.7.	RELAY .....	44
11.8.	MOTOR ARUS SEARAH .....	46
11.8.1	PENGOPERASIAN DAN KARAKTERISTIK MOTOR DC .....	48
11.6.2	PENGATURAN KECEPATAN MOTOR DC SHUNT .....	50

III. PERENCANAAN .....	53
III.1. PENDAHULUAN .....	53
III.2. BLOK DIAGRAM SISTEM .....	53
III.3. PRINSIP KERJA SISTEM .....	54
III.3.1. BAGIAN SISTEM KONTROLLER ....	57
III.3.2. BAGIAN INTERFACE UNTUK IBM PC	58
III.3.3. BAGIAN DRIVER DAN SENSOR ....	59
III.4. PERENCANAAN SISTEM KONTROLLER .....	59
III.4.1. PERLUASAN PROGRAM MEMORI ....	60
III.4.2. PERLUASAN DATA MEMORI .....	62
III.4.3. INPUT - OUPUT PORT .....	64
III.5. TRANSISTOR SEBAGAI PENGEMUDI RELAY ...	66
III.6. INTERFACING PENYIMPANAN MODEL DI-RAM .	68
III.6.1. I/O MAPPING PADA IBM PC .....	69
III.6.2. SISTEM PENGKODEAN .....	70
III.6.3. PENYANGGA .....	71
III.6.4. LATCH .....	73
III.7 PERANGKAT LUNAK .....	75
IV. PENGUKURAN .....	83
IV.1. PENDAHULUAN .....	83
IV.2. PENGUKURAN SINYAL PSEN . ....	83
IV.3. PENGUKURAN SINYAL ALE .....	84
IV.4. CONTOH PEMBUATAN MODEL .....	85
V. PENUTUP .....	91
DAFTAR PUSTAKA .....	93



LAMPIRAN A : DATA SHEET .....	95
LAMPIRAN B : SOFTWARE LISTING .....	102
USULAN TUGAS AKHIR .....	121

## DAFTAR GAMBAR

GAMBAR	HALAMAN
2.1 GERAKAN KETIKA PEMBUBUTAN .....	6
2.2 PROGRAM MEMORY MAP .....	11
2.3 DATA MEMORY MAP .....	12
2.4 STRUKTUR 'QUASI BIDIRECTIONAL PORT' ....	14
2.5 PROGRAM COUNTER STACK .....	17
2.6 PROGRAM STATUS WORD (PSW) .....	19
2.7 INTERRUPT LOGIC .....	20
2.8 TIMER/ EVENT COUNTER .....	23
2.9 CYCLE PEWAKTU .....	25
2.10 EKSTERNAL RESET .....	26
2.11 DIAGRAM WAKTU PENGAMBILAN DATA DARI MEMORY	32
2.12 DIAGRAM WAKTU PENGIRIMAN DATA KE MEMORY ..	33
2.13 KONFIGURASI PIN PADA SLOT IBM PC .....	36
2.14 PEMBAGIAN ALAMAT I/O PORT PADA IBM PC ....	39
2.15 SIMBOL OPTOCOUPLER .....	43
2.16 RELAY HISTERISIS .....	45
2.17 SIMBOL MOTOR DC .....	48
2.18 PENGATURAN TEGANGAN DAN ARUS PADA KUMPARAN	
MEDAN .....	50
2.19 PENGATURAN TEGANGAN DAN ARUS PADA ARMATURE	51

3.1	BLOK DIAGRAM SISTEM .....	54
3.2	PENGAMBILAN INSTRUKSI DARI PROGRAM MEMORY LUAR .....	60
3.3	RANGKAIAN PROGRAM MEMORY LUAR .....	61
3.4	TIMING DATA MEMORY LUAR .....	62
3.5	RANGKAIAN SISTEM DATA MEMORY LUAR ...	63
3.6	PORT 1 SEBAGAI INPUT PORT .....	65
3.7	INTERUPT DAN COUNTER .....	66
3.8	TRANSISTOR SEBAGAI PENGGERAK RELAY ..	67
3.9	ALAMAT I/O PORT PADA SLOT .....	70
3.10	RANGKAIAN SISTEM PENGKODEAN .....	71
3.11	RANGKAIAN BUFFER .....	73
3.12	RANGKAIAN LATCH .....	74
3.13	DIAGRAM ALIR KONTROL MESIN BUBUT ..	78
4.1	BENTUK GELOMBANG PSEN .....	84
4.2	BENTUK SINYAL ALE .....	85
4.3	FOTO DARI ALAT YANG DIBUAT .....	86
4.4	TAMPILAN UTAMA DARI PEMBUATAN MODEL .....	86
4.5	PEMILIHAN GAMBAR MODEL PERTAMA .....	87
4.6	PENGISIAN PARAMETER MODEL PERTAMA ...	87
4.7	PEMILIHAN GAMBAR MODEL KEDUA .....	88
4.8	PENGISIAN PARAMETER MODEL KEDUA .....	88
4.9	PEMILIHAN GAMBAR MODEL KETIGA .....	89
4.10	PENGISIAN PARAMETER MODEL KETIGA ....	89
4.11	HASIL AKHIR DARI MODEL YANG DIBUAT ..	90

## DAFTAR TABEL

TABEL	HALAMAN
2.1 FUNGSI DARI PIN-PIN M8035L .....	28
2.2 I/O PORT DALAM SISTEM BOARD .....	40
2.3 I/O PORT DILUAR SISTEM BOARD .....	41
3.1 TABEL KEBENARAN DECODER 74LS138 .....	71
3.2 TABEL KEBENARAN 74LS245 .....	72
3.3 TABEL KEBENARAN LATCH 74LS377 .....	74

# BAB I

## PENDAHULUAN

---

### 1.1 LATAR BELAKANG

Dalam bidang industri sering dibutuhkan benda kerja presisi dengan bentuk tertentu. Yang ukurannya, baik diameter maupun panjangnya diluar ketentuan yang ada dipasaran.

Untuk menghasilkan benda-benda kerja tersebut, maka salah satu peralatan yang sering digunakan adalah mesin bubut. Mesin bubut digunakan untuk menghasilkan benda kerja berbentuk silinder seperti baut, ulir, as, pegangan tangan dan lain-lain.

Pengerjaan pembubutan dilakukan dengan memutar bahan kerja pada porosnya, dan kemudian pahat akan memakan permukaan bahan kerja tersebut sehingga didapatkan benda kerja sesuai dengan keinginan.

Dengan kemajuan bidang elektronika yang demikian pesat dan ditemukannya mikroprosesor yang menjadi pusat pengolahan data, maka tidak bisa dihindari digunakannya komponen-komponen elektronika tersebut sebagai pengontrol posisi dari pahat, yang mana pada beberapa jenis mesin bubut model lama, pengontrolannya masih dilakukan secara manual.

Dalam tugas akhir ini akan dibahas mengenai

## BAB I PENDAHULUAN

---

### 1.1 LATAR BELAKANG

Dalam bidang industri sering dibutuhkan benda kerja presisi dengan bentuk tertentu. Yang ukurannya, baik diameter maupun panjangnya diluar ketentuan yang ada dipasaran.

Untuk menghasilkan benda-benda kerja tersebut, maka salah satu peralatan yang sering digunakan adalah mesin bubut. Mesin bubut digunakan untuk menghasilkan benda kerja berbentuk silinder seperti baut, ulir, as, pegangan tangan dan lain-lain.

Pengerjaan pembubutan dilakukan dengan memutar bahan kerja pada porosnya, dan kemudian pahat akan memakan permukaan bahan kerja tersebut sehingga didapatkan benda kerja sesuai dengan keinginan.

Dengan kemajuan bidang elektronika yang demikian pesat dan ditemukannya mikroprosesor yang menjadi pusat pengolahan data, maka tidak bisa dihindari digunakannya komponen-komponen elektronika tersebut sebagai pengontrol posisi dari pahat, yang mana pada beberapa jenis mesin bubut model lama, pengontrolannya masih dilakukan secara manual.

Dalam tugas akhir ini akan dibahas mengenai

perencanaan dan pembuatan pengontrol kerja dari mesin bubut, yaitu pengontrolan dari posisi pahat dan gerakan pemakanan dengan menggunakan mikrokomputer sebagai sarana pengontrol kerja dari mesin tersebut.

## 1.2 PERMASALAHAN

Dalam prakteknya, untuk mendapatkan diameter benda kerja yang diinginkan dari proses pembubutan, adalah suatu proses yang memerlukan ketelitian dan ketelatenan cukup tinggi. Begitu juga jika diinginkan suatu pekerjaan yang masal, dimana sebuah model digunakan berkali-kali.

Untuk mengatasi permasalahan tersebut maka digunakan kontroller sebagai pengatur posisi pahat dan pengatur gerakan pemakanan. Dan sebagai penyimpan data model digunakan RAM, sehingga sebuah model dapat digunakan untuk proses yang berulang.

## 1.3 PEMBATAHAN MASALAH

Untuk perencanaan pembuatan pengontrol kerja mesin bubut ini panjang daerah pemakanan dibatasi sampai 250 mm dan panjang daerah pengaturan posisi pahat dibatasi sampai 50 mm.

Untuk pembuatan gambar model yang dilakukan oleh IBM PC/XT digunakan program bahasa pascal, sedang untuk pengolahan yang dilakukan oleh kontroller bahasa

program yang digunakan adalah assembly 8035.

#### 1.4 METODOLOGI

Gambar model yang dibuat di IBM PC/XT ditransfer ke RAM melalui card interface. Informasi model yang dikirim ke-RAM adalah besar sumbu y setiap perubahan 1 mm dari panjang sumbu x. Selanjutnya data yang ada di-RAM akan dibandingkan oleh kontroller dengan posisi dari pahat, jika data tidak sama dengan posisi pahat maka kontroller akan menggerakkan motor dc sampai posisi sama dengan data yang ada. Jika posisi telah sama dengan data maka motor pemakanan akan maju 1 mm dan kontroller akan mengambil data selanjutnya untuk dibandingkan dengan posisi yang baru dan begitu seterusnya.

#### 1.5 LANGKAH LANGKAH PEMBAHASAN

Setelah mengulas tentang bab pendahuluan, pada bab dua akan dibahas segala sesuatu mengenai teori dasar baik mengenai kontroller, teknik menghubungkan perangkat keras ke IBM PC/XT, motor dc dan drivernya dan optocoupler sebagai sensor posisi.

Pada bab ketiga akan dibahas mengenai perencanaan software dan hardwarenya, yaitu mengenai sistem kerja kontroller, komputer dan interfacenya.



Pada bab keempat akan dibahas mengenai hasil pengukuran dari percobaan dan akan dibandingkan dengan data model.

Pada bab yang terakhir akan diberikan kesimpulan menyeluruh sistem yang direncanakan.

## BAB II TEORI PENUNJANG

### II.1 PENDAHULUAN

Pada bab ini akan dibahas teori penunjang yang berhubungan dengan perencanaan alat yang dibuat, pertama akan dibahas sedikit teori mengenai mesin bubut kemudian mengenai komponen serpih tunggal mikrokomputer intel M8035L yang berfungsi sebagai pusat pengontrol, EPROM sebagai sarana penyimpanan program memori, RAM sebagai penyimpan data memori, opto coupler, transistor serta interface ke personal computer .

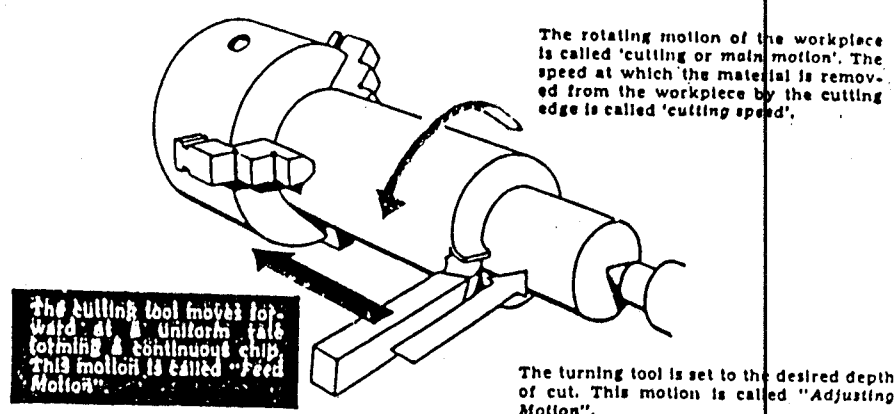
### II.2 MESIN BUBUT<sup>1)</sup>

Mesin bubut adalah peralatan yang digunakan untuk menghasilkan benda kerja berbentuk silinder. Banyak jenis mesin bubut yang digunakan dalam bidang industri, dan yang sering digunakan adalah mesin bubut jenis 'centre lathe'. Proses pembubutan dilakukan dengan jalan memutar bahan kerja pada porosnya. Gerakan

---

<sup>1)</sup> Heinrich Gerling, "All About Machine Tool", Wiley Eastern Limited, New Delhi, 1965, hal 13.

yang dilakukan oleh mesin bubut untuk memutar bahan kerja pada porosnya ini disebut gerakan utama. Pahat kemudian diatur sesuai dengan kedalaman pemotongan yang diinginkan. Gerakan ini disebut gerakan pengaturan. Setelah pengaturan ini dicapai sesuai dengan yang diinginkan maka pahat akan digerakkan kedepan untuk melakukan proses pemakanan. Gerakan ini disebut gerakan pemakanan. Gambar 2.1 menunjukkan jenis-jenis gerakan dari proses pembubutan tersebut.



Gambar 2.1<sup>2)</sup>

Gerakan ketika pembubutan.

### 11.2.1 PAHAT MESIN BUBUT<sup>3)</sup>

Untuk memakan bahan kerja, pahat harus terbuat dari bahan yang mempunyai sifat keras, liat dan tahan

<sup>2)</sup> ibid hal.13

<sup>3)</sup> ibid hal.25

panas. Sifat Keras diperlukan supaya ujung dari pahat dapat menembus kedalam bahan kerja. Pahat yang mempunyai sifat ketiatan yang rendah menyebabkan ujung pahat mudah patah. Sifat tahan panas menjaga kekerasan dari ujung pahat yang menjadi panas ketika pahat bergesekan dengan bahan kerja.

Banyak jenis pahat yang dibedakan menurut bahan pembuatannya. Pahat dari baja tanpa campuran. Dengan campuran karbon 0.5 - 1.5 %. Kekerasannya hilang pada suhu  $250^{\circ}\text{C}$ . Pahat dari baja campuran. Yang terdiri campuran karbon, tungsten, kromium, vanadium dll. Jenis ini hilang kekerasannya pada suhu  $600^{\circ}\text{C}$ . Cemented carbide. Yang terdiri dari campuran tungsten atau molibdenium, karbon dan kobalt, pada suhu  $900^{\circ}\text{C}$  masih bekerja dengan baik. Jenis lain dari pahat adalah pahat dengan ujung berlian yang mempunyai sifat sangat keras. Mereka biasanya digunakan pada tahap penyelesaian dari proses dengan menggunakan mesin khusus.

### 11.3 MIKROKOMPUTER SERPIH TUNGGAL INTEL 8035<sup>4)</sup>

Mikrokomputer Intel M8035L adalah sebuah mikrokomputer 8 bit yang berada dalam satu kemasan, terbuat dari chip silicon tipe N dengan memakai proses

---

<sup>4)</sup> ...., Single-Chip 8-Bit Microcontroller., Philips Electronic Componen Materials., 1986. hal. 2-1

MOS. Mikrokomputer ini terdiri dari :

- 8 bit Unit Pusat Pengolah (CPU)
- 64 x 8 RAM data memori
- 27 jalur Input-Output
- 8 bit pewaktu/ pencacah
- rangkaian osilator dan clock

Mikroprosesor dan *arithmetic processor* dirancang khusus agar mikrokomputer ini mudah dan efisien dipakai sebagai pengontrol .

Program memori yang dibutuhkan cukup efisien karena semua instruksi hanya terdiri dari satu dan dua byte instruksi, tidak ada instruksi yang lebih panjang dari dua byte.

#### 11.3.1. ARSITEKTUR<sup>5)</sup>

##### Bagian aritmatik

Bagian aritmatik dari prosesor fungsi dasar manipulasi data dari M8035L dan dapat dibagi dalam beberapa blok :

- *Arithmetic Logic Unit (ALU)*
- Akumulator
- *Carry Flag*
- Pengkode Instruksi

Cara pengoperasiannya yaitu data yang disimpan didalam

---

<sup>5)</sup>ibid

akumulator dikombinasikan didalam *Arithmetic Logic Unit (ALU)* dengan data yang berasal dari sumber lain misalnya register atau I/O port, dan hasilnya akan disimpan di akumulator atau di register yang lain.

#### Pengkode Instruksi

Kode operasi atau op-code untuk setiap instruksi program disimpan di pengkode instruksi dan dikonversikan ke output yang mengontrol fungsi dari masing-masing blok diagram dari bagian aritmatik. Jalur ini mengontrol sumber dari data dan register tujuan sama baiknya dengan fungsi yang dilakukan di ALU.

#### *Arithmetic Logic Unit*

*Arithmetic Logic Unit (ALU)* menerima 8 bit data yang berasal dari satu atau dua sumber dan dibawah pengaturan dari pengkode instruksi akan menghasilkan kembali data 8 bit.

ALU dapat menjalankan instruksi-instruksi :

- *Add With or Without Carry*
- *AND, OR, Exclusive OR*
- *Increment/ decrement*
- *Swap Nibbles*
- *BCD Decimal Adjust*

bila operasi yang dihasilkan ALU lebih besar dari 8 bit (overflow of most significant bit), maka *carry flag*

pada program status word akan di set.

### Akumulator

Akumulator adalah sebuah register yang terpenting didalam prosesor, menjadi salah satu input yang akan dioperasikan didalam ALU serta seringkali menjadi tempat hasil setelah operasi.

Data yang berasal dari dan yang menuju I/O port, normalnya juga melalui akumulator.

### 11.3.2 PROGRAM MEMORI<sup>6)</sup>

Mikrokomputer M8035L ini tidak mempunyai internal program memori, oleh karena itu kita harus menambahkan program memori luar.

Didalam program memori ini terdapat tiga lokasi penting seperti terlihat pada gambar 2.2. yaitu:

#### - Lokasi 0

Pengaktifan jalur reset dari prosessor akan menyebabkan instruksi pertama yang akan diambil dari lokasi 0.

#### - Lokasi 3

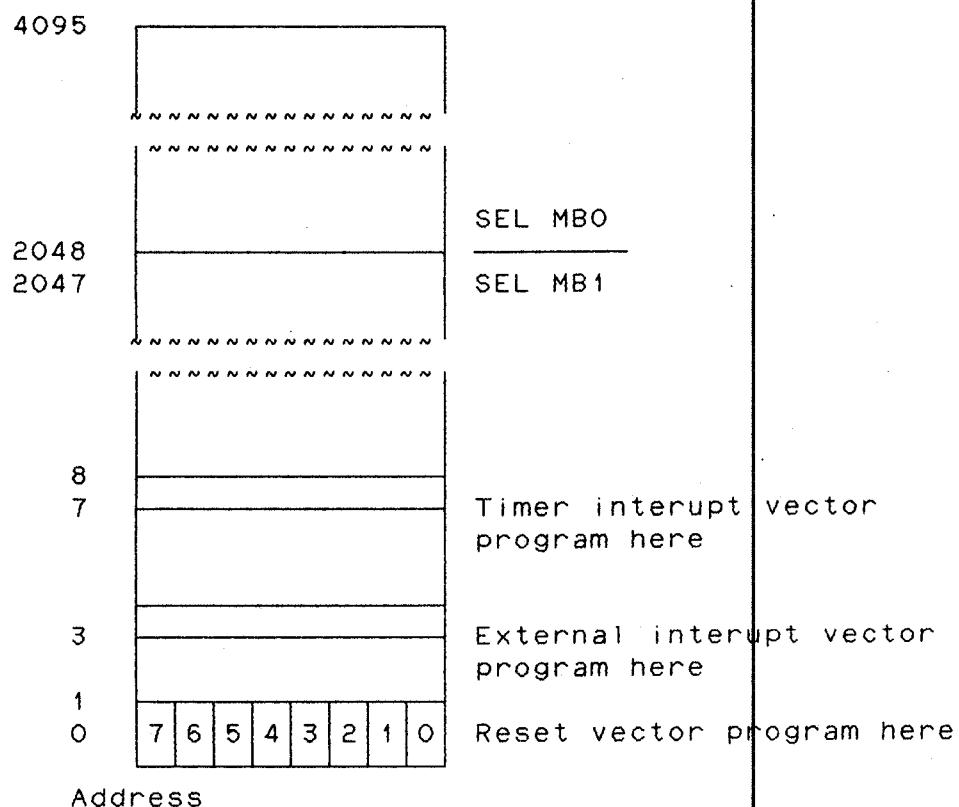
Pengaktifan jalur input interrupt dari prosessor (jika interrupt di-enable) menyebabkan meloncat ke subrutin pada lokasi 3.

---

<sup>6)</sup>ibid

- Lokasi 7

Timer/ counter interrupt yang disebabkan oleh adanya overflow pada timer/ counter (jika di-enable) menyebabkan melompat ke subrutin di lokasi 7.



Gambar 2.2

Program memory map

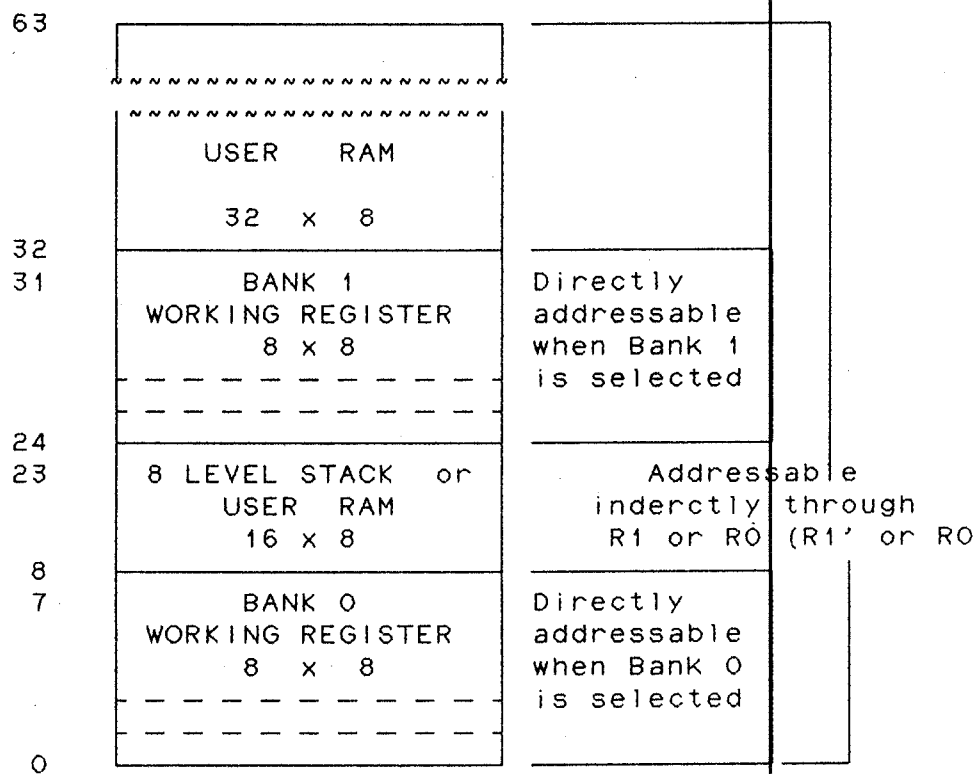
### 11.3.3 DATA MEMORI<sup>7)</sup>

Didalam chip mikrokomputer ini terdapat data

<sup>7)</sup> ibid hal.2-3



memori sebanyak  $64 \times 8$  bit. Setiap lokasi bisa dihubungi secara langsung dengan menggunakan dua buah RAM pointer yang disimpan pada alamat 0 dan 1 dari register array. Gambar 2.3 menunjukkan data memori map.



Gambar 2.3

#### Data memory Map

Lokasi 0 sampai dengan 7 digunakan sebagai register kerja. Bila kita menjalankan instruksi Register Bank switch (SEL RB1), lokasi 24 sampai 31 menjadi lokasi register kerja pada Bank 1, yaitu sama dengan lokasi 0 sampai dengan lokasi 7 pada Bank 0.

Bank kedua dari lokasi kerja register dapat digunakan sebagai perluasan dari bank pertama atau

dipakai sebagai cadangan selama menjalankan subrutin interrupt. register pada Bank 0 biasanya digunakan untuk program utama dan bila terjadi interrupt bisa segera disimpan dengan menggunakan Bank Switch. Bila kita tidak melaksanakan instruksi SEL RB, maka bank kedua lokasi 24 sampai dengan 31 bisa digunakan untuk keperluan pemakai.

Sedangkan lokasi 8 sampai dengan 23 dipakai sebagai *Program Counter stack*. Lokasi ini langsung bisa dihubungi oleh Stack Pointer selama sedang melaksanakan subrutin calls.

#### 11.3.4 INPUT / OUTPUT<sup>8)</sup>

Mikrokomputer M8035L mempunyai 27 jalur I/O yang dapat dipakai sebagai input atau output. Jalur jalur ini dikelompokkan sebagai 3 port yang terdiri dari 8 jalur yang masing masing berlaku sebagai jalur input, output atau bidirectional port dan 3 'test input'.

##### a. Port 1 dan Port 2

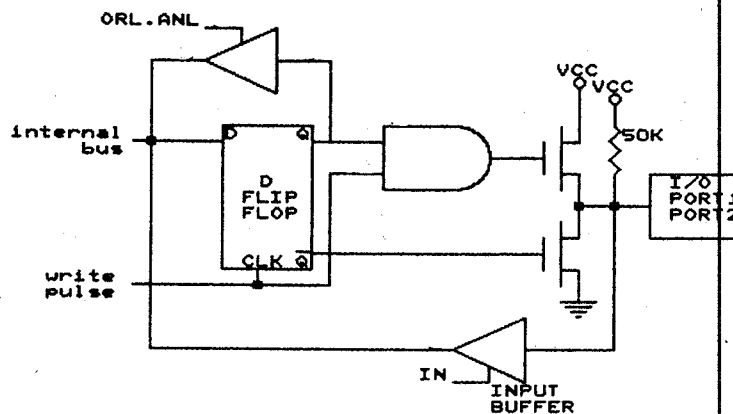
Masing-masing mempunyai panjang 8 bit dan keduanya mempunyai karakter yang sama, data yang ditulis ke port ini dapat dipertahankan nilainya (latched) sampai port tersebut ditulis kembali. Akan tetapi input port tidak

---

<sup>8)</sup> ibid hal. 2-4

latch, jadi input harus selalu ada sampai terjadi proses pembacaan.

Port 1 dan port 2 disebut quasi bidirectional disebabkan terdapat struktur rangkaian khusus yang memungkinkan setiap jalur dapat berfungsi sebagai input, output atau bahkan keduanya. Gambar 2.4 menunjukkan struktur port quasi bidirectional.



Gambar 2.4

Struktur 'quasi bidirectional port'

Masing-masing jalur selalu dipull-up pada 5 Volt melalui sebuah resistor yang mempunyai impedansi relatif tinggi ( $\sim 50 \text{ K}\Omega$ ). Pull up ini sudah cukup memberikan sumber arus TTL pada level tinggi dan dapat diturunkan ke-low (pulled low) oleh sebuah gerbang TTL standart sehingga memungkinkan pin yang sama digunakan sebagai input dan output. Untuk menghasilkan switching yang cepat dari transisi "0" ke "1", sebuah komponen dengan impedansi yang relative rendah ( $\sim 5 \text{ K}\Omega$ ) diswitch

secara cepat (~500 ns) apabila '1' dituliskan ke jalur. Reset menginisialisasi semua jalur ini pada keadaan '1'. Keadaan ini mengijinkan input dan output pada pin yang sama dan juga mengijinkan percampuran dari jalur input dan jalur output pada port yang sama.

Port quasi bidirectional ini dengan kombinasi instruksi logika ANL dan ORL menghasilkan keadaan yang efisien untuk mengatasi masalah sebuah jalur digunakan sebagai input dan output.

#### b. BUS

Bus juga merupakan port 8 bit bidirectional port. Jika keadaan bidirectional tidak diperlukan, BUS dapat melayani sebagai output port yang dilatch ataupun sebagai input port. Jalur-jalur input dan output pada port ini tidak dapat dicampur. Sebagai port, data ditulis dan dilatch dengan menggunakan instruksi OUTL dan di-input-kan dengan menggunakan instruksi INS. Instruksi INS dan OUTL menghasilkan pulsa pada jalur  $\overline{RD}$  dan  $\overline{WR}$ . Sebagai bidirectional port, instruksi MOVX yang biasa digunakan untuk membaca dan menulis port. Menulis ke port menghasilkan sebuah pulsa pada jalur  $\overline{WR}$  dan data valid pada saat pulsa turun dari  $\overline{WR}$ . Pembacaan dari port menghasilkan sebuah pulsa pada jalur  $\overline{RD}$  dan data input harus valid pada saat pulsa turun dari  $\overline{RD}$ . Pada saat tidak membaca atau menulis jalur BUS ini dalam keadaan impedansi tinggi.

#### 11.3.4 TEST DAN INT INPUT<sup>9)</sup>

Tiga buah pin disediakan untuk input sekaligus berfungsi sebagai test dengan instruksi conditional jump, ketiga kaki tersebut adalah T0, T1 dan INT. Fungsi masing-masing pin ini akan dijelaskan pada bagian yang membahas masalah interrupt, counter dan timer.

#### 11.3.5 PROGRAM COUNTER DAN STACK<sup>10)</sup>

*Program counter* adalah counter yang berdiri sendiri, sedangkan *program counter stack* adalah peralatan yang menggunakan sepasang register didalam area data memory. Program counter ini diisi dengan 0 pada saat reset awal.

Dengan adanya interrupt atau perintah Call ke subrutin, menyebabkan isi daripada program counter disimpan didalam satu dari delapan pasangan register pada program counter stack, seperti ditunjukkan pada gambar 2.5. Pasangan yang digunakan ditentukan oleh 3 bit stack pointer yang merupakan bagian dari *Program Status Word (PSW)* seperti ditunjukkan pada gambar. Data RAM lokasi 8 sampai 23 digunakan sebagai stack register dan digunakan untuk menyimpan program counter

---

<sup>9)</sup>ibid

<sup>10)</sup>ibid

dan 4 bit dari PSW seperti yang ditunjukkan pada gambar. Stack pointer ketika di-inisialisasi ke 000 akan menunjuk RAM lokasi 8 dan 9. Subrutin pertama akan memindahkan isi program counter ke lokasi 8 dan 9 dari RAM array, setelah itu stack pointer dinaikkan satu menunjuk ke RAM lokasi 10 dan 11 untuk menangani CALL yang lain.

POINTER		
111		R23
		22
110		21
		20
101		19
		18
100		17
		16
011		15
		14
010		13
		12
001		11
		10
000	PSW	9
	PC4-7	R 8
	PC0-3	
	MSB	LSB

Gambar 2.5

Program Counter Stack

Sekumpulan subrutin didalam subrutin dapat dijalankan sampai delapan kali tanpa bertumpukan dengan yang lain pada stack. Jika overflow terjadi pada alamat terdalam (lokasi 8 dan 9), maka terjadi penulisan ulang yang menyebabkan hilangnya program counter pada stack sebelumnya.

Berakhirnya subrutin ditandai dengan adanya instruksi return yaitu RET atau RETR. Instruksi ini menyebabkan Stack Pointer diturunkan nilainya dengan satu dan pasangan register ini dipindahkan ke-program counter.

#### 11.3.6 PROGRAM STATUS WORD<sup>(1)</sup>

Delapan bit status word yang dapat diambil atau diberikan oleh akumulator, disebut *Program status Word*. Program status word sebenarnya merupakan sekumpulan dari flip-flop yang dapat dibaca atau ditulis secara keseluruhan. Kemampuan untuk menulis ke program status word memberikan kemudahan untuk memperbaiki status mesin bila terjadi power down. Empat bit bagian atas dari PSW disimpan di Program Counter Stack.

Empat bit teratas dari PSW disimpan diprogram counter stack pada saat pemanggilan subrutin atau interrupt, dan akan dikeluarkan kembali setelah instruksi RETR.

---

<sup>(1)</sup> ibid hal. 2-5

Program Status Word bit didefinisikan sebagai berikut :

Bit 0-2 : bit-bit stack pointer (S0, S1, S2)

Bit 3 : Tidak digunakan (bila sedang membaca berada pada level "1")

Bit 4 : Working Register Bank Switch bit (BS)

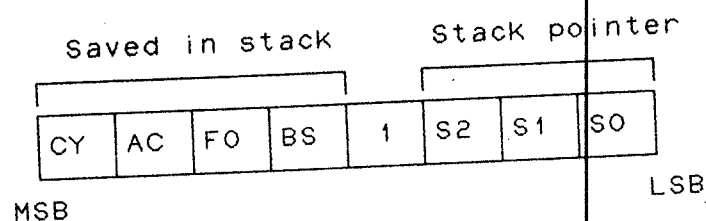
0 = Bank 0

1 = Bank 1

Bit 5 : Flag 0 bit (F0) digunakan untuk mengontrol flag, dapat dikomplemenkan atau dikosongkan dan merupakan test pada instruksi conditional jump JF0.

Bit 6 : Auxilliary Carry (AY) carry bit yang dihasilkan pada instruksi ADD dan digunakan pada instruksi decimal adjust DA A.

Bit 7 : Carry (CY) carry flag, berguna untuk memberikan tanda pada operasi selanjutnya bahwa hasil di akumulator terjadi overflow.



CY = Carry

AC = Auxilliary Carry

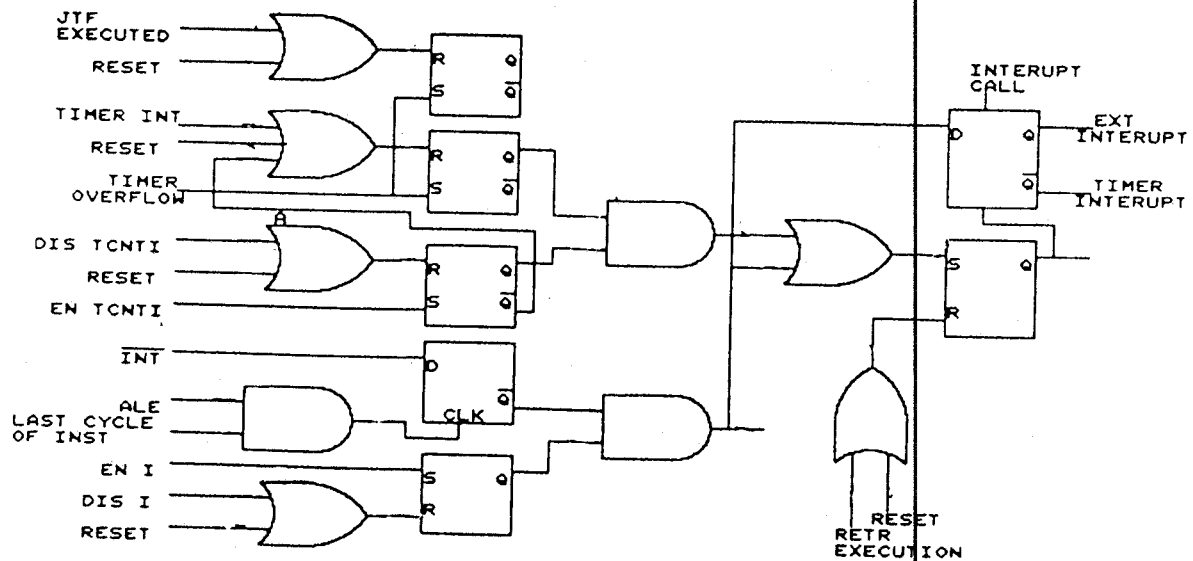
F0 = Flag 0

BS = Register Bank select

Gambar 2.6

Program Status Word (PSW)



11.3.8 INTERRUPT<sup>13)</sup>

Gambar 2.7

## interrupt logic

Urutan interrupt dimulai dengan memberikan input level rendah ke pin INT. Gambar 2.7 menunjukkan logika interrupt dari M8035L. Jalur interrupt diambil setiap cycle mesin dari instruksi cycle tunggal selama ALE dan apabila dideteksi akan menyebabkan melompat kesubrutin pada lokasi 3 pada program memori secepat seluruh cycle dari instruksi pada saat itu selesai. Pada instruksi dua cycle jalur interrupt diambil oleh ALE hanya pada cycle kedua. INT harus ditahan pada logika rendah paling sedikit 3 Tcy untuk meyakinkan bahwa operasi

<sup>13)</sup> ibid hal.2-7

paling sedikit 3 Tcy untuk meyakinkan bahwa operasi interrupt sedang dilakukan. Seperti pada CALL ke subrutin, Program Counter dan Program Status Word disimpan pada stack. Memori program lokasi 3 biasanya berisi jump tak bersyarat ke subrutin yang melayani interrupt. Akhir dari subrutin tersebut diketahui dengan memberikan instruksi RETR. Jika internal timer/ counter menghasilkan interrupt dan pada saat yang sama terjadi external interrupt, maka yang akan dikenali adalah external interrupt.

#### Interrupt Timing

Interrupt input dapat diaktifkan dan di non-aktifkan dibawah Program Control dengan menggunakan instruksi EN I atau DIS I. Interrupt dinon-aktifkan oleh reset dan tetap dalam keadaan tersebut sampai diaktifkan oleh program pemakai. Permintaan interrupt harus dihilangkan sebelum instruksi RTRN dijalankan selama kembali dari pelayanan rutin, jika tidak prosessor akan segera memasukkan pelayanan rutin interrupt lagi.

#### 11.3.9 Timer/ Counter<sup>14)</sup>

M8035L berisi counter yang membantu pemakai untuk menghitung kejadian luar (*external event*) dan menghasilkan waktu tunda yang tepat tanpa meletakkan

---

<sup>14)</sup> ibid hal.2-8

sebuah rangkaian tambahan pada prosessor untuk keperluan ini. Pada kedua mode ini, counter mempunyai operasi yang sama. Perbedaannya hanya adanya sumber input untuk counter.

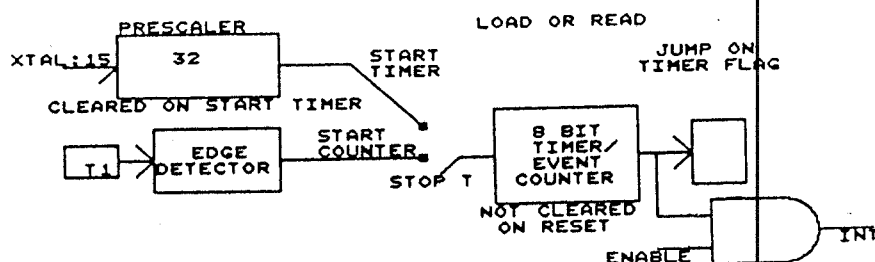
### Counter

8 bit binari counter dapat di-preset dan dibaca dengan dua instruksi MOV, yang memindahkan isi dari akumulator ke-counter dan sebaliknya. Counter akan dihentikan oleh Reset atau instruksi STOP TCNT dan akan tetap dihentikan sampai ada instruksi START T jika berlaku sebagai timer, atau instruksi START CNT jika berlaku sebagai counter. Sekali diaktifkan counter akan menaikkan hitungan sampai maksimum (FFh) dan overflow ke nol untuk meneruskan hitungannya, sampai dihentikan oleh instruksi STOP TCNT atau reset. Increment dari hitungan maksimum ke nol (overflow) menyebabkan di-set-nya flag overflow. Keadaan dari flag overflow dapat di-test dengan instruksi lompat bersyarat JTF. Flag direset dengan menjalankan JTF atau oleh reset.

Permintaan interrupt disimpan didalam latch dan kemudian di-OR-kan dengan input interrupt luar INT. Timer interrupt dapat di-enable dan disable secara bebas dari interrupt luar dengan instruksi EN TCNTI dan DIS TCTNI. Jika di-enable, counter overflow akan menyebabkan pamanggilan subrutin kelokasi 7 dimana pelayanan rutin timer atau counter disimpan. Jika timer dan interrupt

luar terjadi secara bersamaan, interrupt luar akan dikenali dulu dan akan menuju lokasi 3, timer interrupt akan dilatch dan tetap dalam keadaan tersebut sampai pelayanan interrupt luar kembali dari subrutin. Timer interrupt direset dengan pemanggilan kelokasi 7 atau dapat dilakukan dengan menjalankan instruksi DIS TCNT1.

#### 11.3.9.1 Sebagai Pencacah



Gambar 2.8

Pewaktu/ pencacah

Instruksi START CNT akan menghubungkan pin input T1 ke input pencacah dan mengaktifkan pencacah. T1 input di-sample pada awal dari keadaan 3. Perpindahan dari logika tinggi ke rendah pada T1 akan menyebabkan pencacah bertambah. T1 harus ditahan sekurang-kurangnya 1 Tcy untuk memastikan pencacah diterima. Harga maksimum sehingga pencacah masih dapat bertambah adalah sepertiga cycle instruksi (setiap 5,7  $\mu$ sec apabila menggunakan kristal 8 Mhz) - tidak ada harga frekuensi minimum. Input T1 harus tetap tinggi paling sedikit 1/5 dari Tcy setelah setiap transisi.

setiap transisi.

### Timer

Instruksi START T akan menghubungkan internal clock ke input counter dan mengaktifkan counter. Internal clock diperoleh dengan melewatkan clock ALE melalui pembagi 32. Pembagi direset selama instruksi START T. Hasil clock 16.6 KHz akan menambah counter setiap 60 usec. Delay antara 60 usec dan 15 msec (256 hitungan) dapat dihasilkan dengan preset counter dan mendeteksi overflow. Untuk delay lebih dari 15 msec dapat dihasilkan dengan menjumlahkan beberapa kali overflow di register dibawah pengontrolan software. Untuk resolusi waktu yang kurang dari 60 usec clock luar dapat digunakan pada input T1 dan counter dioperasikan sebagai mode event counter. ALE yang dibagi 3 atau lebih dapat melayani sebagai clock luar untuk keperluan tersebut.

### 11.3.10 CLOCK DAN RANGKAIAN PEWAKTU<sup>15)</sup>

#### Osilator

Osilator adalah sebuah rangkaian resonan paralel dengan gain yang tinggi, yang mempunyai daerah frekuensi dari 1 sampai 8 Mhz. Pin luar X1 adalah input ke tahap penguatan, sedang X2 adalah output. Kristal

---

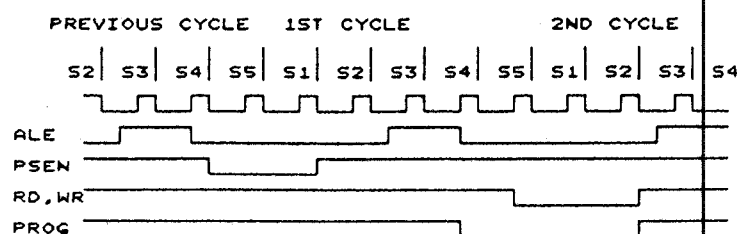
<sup>15)</sup> ibid hal. 2-9

atau induktor yang dihubungkan antara X1 dan X2 menghasilkan umpan balik dan pergeseran phase yang diperlukan untuk osilasi. Jika ketepatan dan kecepatan maksimum prosessor tidak begitu diperlukan, maka induktor dapat digunakan untuk menggantikan kristal. Dengan induktor, frekuensi osilator kira-kira dari 3 sampai 5 Mhz. Untuk pengoperasian dengan kecepatan yang lebih tinggi maka harus menggunakan kristal.

#### 11.3.10.2 State Counter

Output dari osillator dibagi 3 dalam state counter untuk menghasilkan clock yang menentukan keadaan waktu dari mesin (CLK). CLK dapat digunakan pada eksternal pin T0 dengan menjalankan instruksi ENTO CLK. Output dari CLK pada T0 dinon-aktifkan dengan reset dari prosessor.

#### 11.3.10.3 Cycle Counter

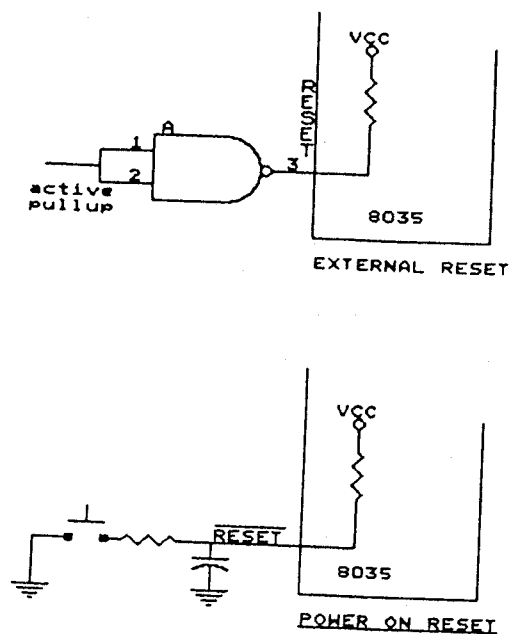


Gambar 2.9

cycle counter

CLK kemudian dibagi 5 pada *cycle counter* untuk menghasilkan clock yang menentukan sebuah *machine cycle* yang terdiri dari 5 *machine state* seperti ditunjukkan pada gambar 2.9. Clock ini disebut *Address Latch Enable* (ALE). Clock ini dihasilkan secara kontinu pada pin output ALE.

### 11.3.11 RESET<sup>16)</sup>



Gambar 2.10  
reset luar

Input reset digunakan untuk inisialisasi prosessor. Input schmit trigger ini mempunyai resistor pull-up internal yang dalam gabungannya dengan

<sup>16)</sup> ibid hal. 2-9

ditunjukkan pada gambar 2.10.

Jika pulsa reset dihasilkan secara external pin reset harus ditahan pada ground sedikitnya 10 msec.

Reset menghasilkan keadaan-keadaan sebagai berikut:

1. Menempatkan program counter ke nol
2. Menempatkan stack pointer ke nol
3. Memilih register bank nol
4. Memilih memory bank nol
5. Menempatkan BUS pada keadaan impedansi tinggi
6. Menempatkan port 1 dan 2 pada mode input
7. Disable interrupt (timer dan external)
8. Menghentikan timer
9. Clear timer flag
10. Clear FO dan F1
11. Disable clock output dari T0

#### 11.3.12 PENJELASAN PIN-PIN<sup>17)</sup>

Processor 8035 dikemas dalam 40 pin dual in line. Tabel 2.1 merupakan kesimpulan dari fungsi-fungsi dari setiap pin.

---

<sup>17)</sup> ibid hal. 2-15



TABEL 2.1

FUNGSI DARI PIN-PIN 8035

	No. Pin	Fungsi
V <sub>SS</sub>	20	Ground
V <sub>DD</sub>	26	Supply untuk pemrograman +25 Volt
V <sub>CC</sub>	40	Supply utama, +5 Volt
PROG	25	Pulsa program
P10-P17 (Port 1)	27 - 34	8 bit quasi-bidirectional port.
P20-P27 (Port 2)	35 - 38	8 bit quasi bidirectional port P20-P23 4 bit tertinggi program counter selama mengambil program memory dari luar.
DO - D7 (BUS)	12 - 19	Bidirectional port yang dapat ditulis atau dibaca secara serentak dengan RD, WR. Port ini juga bisa dilatch. Merupakan 8 bit terendah program counter selama mengambil program memory dari luar, dan menerima alamat instruksi dibawah kontrol PSEN. Juga berisi alamat dan data dari RAM luar, dibawah kontrol ALE, RD dan WR.
TO	1	Dapat dirancang sebagai clock luar dengan instruksi ENTO CLK. TO juga digunakan selama pemrograman.
T1	39	Dapat dirancang sebagai input counter dengan menggunakan instruksi STRT CNT

## 11.4 MEMORI

Unit memori digunakan untuk menyimpan program yang akan dijalankan oleh CPU. Program tersebut diperlukan untuk mengoperasikan sistem mikrokomputer. Dalam sistem mikrokomputer ini ada dua tipe memori yang digunakan yaitu ROM dan RAM.

### 11.4.1 ROM (READ ONLY MEMORY)<sup>18)</sup>

Memori ini menyimpan program yang permanen dan hanya dapat dibaca saja atau dicopy ke dalam memori, jadi tidak dapat menulis informasi di dalamnya. ROM hanya berisi program yang tidak dapat diubah oleh programmer. Contohnya adalah program monitor suatu mikrokomputer, dimana program ini mengontrol kerja seluruh sistem dan mengizinkan programmer untuk menjalankan sebuah program, memasukkan atau mengeluarkan data, memeriksa atau memodifikasi isi memori dan lain-lainnya. Dedicated program yaitu program yang digunakan untuk melaksanakan fungsi tertentu, misalnya untuk mengontrol kerja mesin dalam suatu proses industri, pengaturan rambu lalu-lintas dan lain-lainnya.

Memori ROM adalah non-volatile yang berarti

---

<sup>18)</sup> James W. Coffron, Z80 Application, Sybex, 1983 hal. 1

bahwa ketika power supply dimatikan maka informasi yang ada didalam ROM tidak akan hilang. Ada beberapa tipe memori non-volatile yang digunakan dalam suatu sistem mikroprosesor yang masing-masing jenis memori mempunyai karakteristik yang sama :

1. ROM (Read Only Memory)
2. PROM (Programable Read Only Memory)
3. EPROM (Erasable Programable Read Only Memory)
4. EARAM (Electrically Alterable Read Only Memory)

#### 11.4.2 RAM (RANDOM ACCESS MEMORY)<sup>19)</sup>

RAM digunakan untuk menyimpan data untuk sementara waktu. Karena penyimpanan data bersifat sementara maka data akan hilang pada saat power dimatikan. RAM termasuk jenis memory volatile. Dalam penyimpanan data ada dua jenis RAM yang digunakan, yaitu :

1. RAM statik
2. RAM dinamik

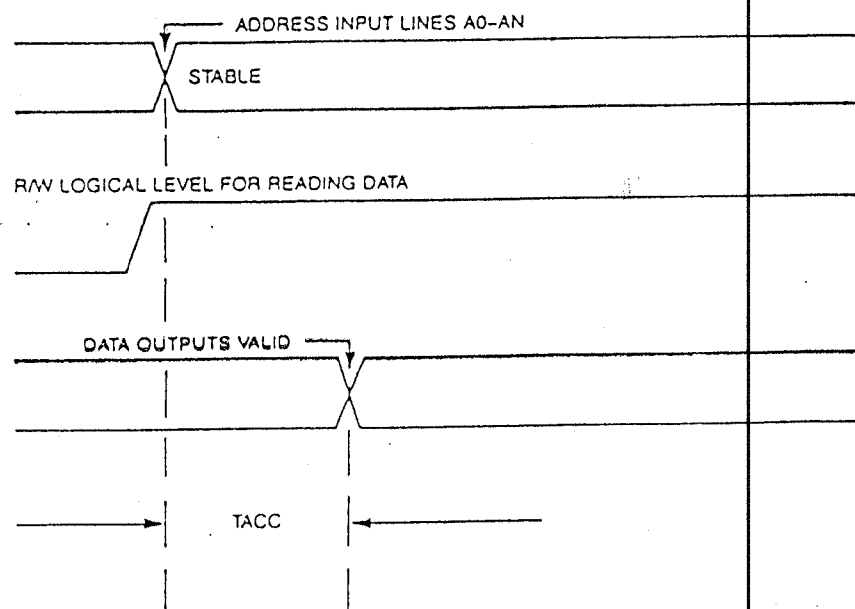
Banyaknya data yang dapat disimpan dalam memory dinyatakan dalam byte.

---

<sup>19)</sup> ibid hal.27

#### 11.4.3 PENGAMBILAN DATA DARI MEMORY<sup>20)</sup>

Diagram waktu pengambilan data dari memory ditunjukkan pada gambar 2.11. Urut-urutan pengambilan data dari memory adalah sebagai berikut :



Gambar 2.11

Diagram waktu pengambilan data dari memory

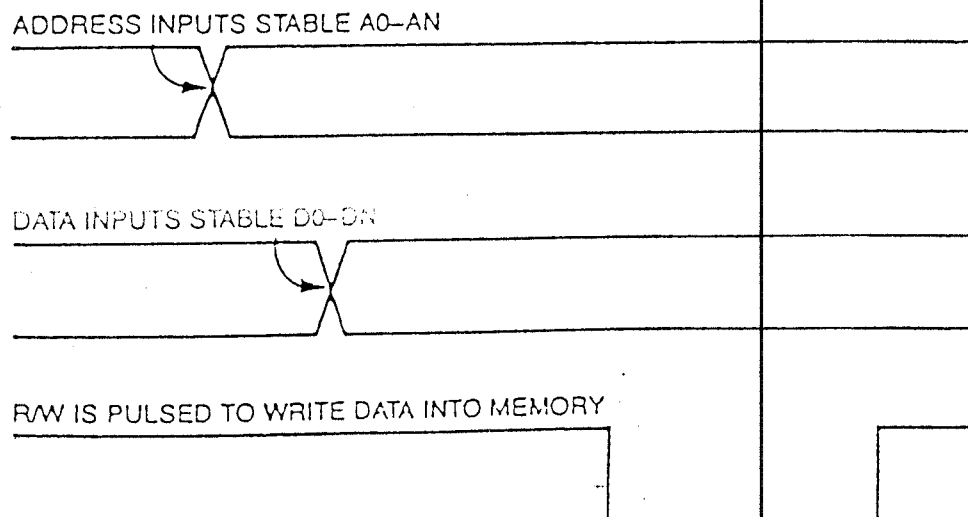
1. Pertama, saluran alamat input diberikan ke memory dan kemudian memory mendekode alamat tersebut dan memberikan lokasi pada data yang akan diambil.
2. Berikutnya, sinyal kontrol CPU ditempatkan dalam kondisi logika yang tepat untuk pengambilan data.

<sup>20)</sup> ibid hal.30

3. Sistem menunggu waktu beberapa saat yang disebut *read acces time* yang akan mengijinkan rangkaian internal dalam memori mendekodekan alamat dan memilih data.
4. Setelah waktu ini data siap ditempatkan pada saluran data bus dan dapat diambil oleh mikro-prosessor.

Proses pengambilan data dalam memory dapat dilakukan baik oleh ROM maupun RAM.

#### 11.3.4 PENGIRIMAN DATA KE MEMORY<sup>21)</sup>



Gambar 2.12

Diagram waktu pengiriman data ke memory

<sup>21)</sup> ibid hal.31

Untuk dapat mengerti proses pengiriman data ke dalam memory dapat dilihat diagram waktu yang ditunjukkan pada gambar 2.12. Secara garis besar urutan pengiriman data kedalam memory diberikan sebagai berikut :

1. Pertama, saluran alamat input diberikan memori pada saat ini lokasi internal dari data yang akan dikirim secara logika didekode oleh memori.
2. selanjutnya, data yang akan dikirim kedalam memory ditempatkan pada saluran data input.
3. Sistem menunggu waktu untuk beberapa saat yang disebut *write access time* agar sistem menjadi stabil.
4. Setelah waktu access ini sinyal kontrol CPU di-set ke-logika yang tepat yang akan menempatkan data kedalam memory.

### 11. 5. INTERFACING KE IBM PC

Untuk membuat model/ gambar dari bahan kerja yang akan dibubut, digunakan mikrokomputer dalam hal ini IBM PC. Selanjutnya data model yang dihasilkan tersebut disimpan dalam RAM, sebagai penyimpan data sementara. Dan selanjutnya data dari RAM tersebut akan diproses oleh mikrokomputer 8035, yang dengan perangkat pembantunya akan melakukan pemakanan terhadap bahan kerja sesuai dengan data yang ada pada

RAM. Untuk menghubungkan IBM PC dengan perangkat luar, dalam hal ini RAM, maka diperlukan unit input/output.

Unit input/output dihubungkan ke IBM PC melalui salah satu slotnya dengan cara yang disebut interfacing. Dalam teknik interfacing, ada dua hal yang selalu dijumpai yaitu penyangga dan pengkode. Penyangga digunakan untuk mengisolasi sistem mikrokomputer dengan peralatan luar pada saat peralatan luar tidak diaktifkan. Pengkode digunakan untuk menghasilkan sinyal pemilih peralatan luar atau yang biasa disebut sinyal *chip select* (CS). Untuk dapat mewujudkan kedua hal tersebut dibutuhkan pengetahuan tentang sistem I/O slot dan I/O port addressing dari IBM PC.

#### 11.5.1 SLOT PERLUASAN IBM PC

Pada IBM PC terdapat 6 buah slot perluasan yang identik. Masing-masing slot mempunyai 62 pin sinyal. Rangkaian interface dapat dipasang disembarang slot yang belum digunakan. Hal ini dimungkinkan karena semua slot tidak dikodekan oleh rangkaian didalam sistem board-nya.

Gambar 2.13 menunjukkan konfigurasi pin-pin pada tiap-tiap slot.

SIGNAL	PIN	PIN	SIGNAL
GND	B1	A1	I/O CH CK
RESET DRV	B2	A2	D7
+5V DC	B3	A3	D6
IRQ2	B4	A4	D5
-5V DC	B5	A5	D4
DRQ2	B6	A6	D3
-12V DC	B7	A7	D2
(NOT USED)	B8	A8	D1
+12V DC	B9	A9	D0
GND	B10	A10	I/O CH RDY
MEMW	B11	A11	AEN
MEMR	B12	A12	A19
IOH	B13	A13	A18
IOR	B14	A14	A17
DACK3	B15	A15	A16
DRQ3	B16	A16	A15
DACK1	B17	A17	A14
DRQ1	B18	A18	A13
DACK0	B19	A19	A12
CLK	B20	A20	A11
IRQ7	B21	A21	A10
IRQ6	B22	A22	A9
IRQ5	B23	A23	A8
IRQ4	B24	A24	A7
IRQ3	B25	A25	A6
DACK2	B26	A26	A5
I/C	B27	A27	A4
ALE	B28	A28	A3
+5V DC	B29	A29	A2
OSC	B30	A30	A1
GND	B31	A31	A0

Gambar 2.13<sup>22)</sup>

Konfigurasi pin pada slot IBM PC

<sup>22)</sup>Lewis C. Eggebrecht, Interfacing To IBM Personal Computer, Indiana Polis : Howard W. Sams & Co., 1987), hal. 77



Secara umum sinyal pada tiap-tiap slot terdiri atas 2 macam sinyal clock, 20 jalur alamat, 8 jalur data, sinyal-sinyal kontrol untuk I/O dan memori, sinyal-sinyal kontrol untuk interrupt, sinyal-sinyal kontrol untuk DMA dan jalur power supply. Berikut akan dijelaskan fungsi dari beberapa jenis sinyal yang digunakan dalam tugas akhir ini.<sup>23)</sup>

- Reset DRV

Reset DRV adalah sinyal yang digunakan untuk mereset sistem board dan semua piranti pendukungnya termasuk untuk mereset peralatan I/O.

- A0 sampai A19

Merupakan jalur sinyal output yang digunakan sebagai jalur alamat yang menunjukkan lokasi memory atau I/O port. Untuk keperluan memory mapping dan I/O port dapat digunakan sebagian atau seluruhnya sebagai input pada pengkode penghasil sinyal pemilih piranti tertentu.

- D0 sampai D7

Merupakan jalur dua arah untuk perpindahan data dari memori atau I/O port ke-sistem board atau sebaliknya.

-  $\overline{\text{IOR}}$

Sinyal ini digunakan untuk menandakan pada I/O bahwa akan dilakukan proses pembacaan oleh CPU.

---

<sup>23)</sup> ibid hal. 53-58

-  $\overline{IOW}$

Sinyal ini digunakan untuk menandakan pada I/O bahwa akan dilakukan proses penulisan data oleh CPU.

- AEN

Merupakan sinyal output dari pengontrol DMA untuk menunjukkan bahwa sedang dilakukan proses DMA. Sinyal ini perlu disertakan pada pengkodean peralatan interface untuk mencegah terjadinya konflik data antara salah satu I/O port yang tidak melakukan proses DMA dengan I/O port yang sedang melakukan operasi DMA. Hal ini mungkin terjadi karena semua bus berada dalam kontrol *DMA controller* pada saat berlangsung operasi DMA.

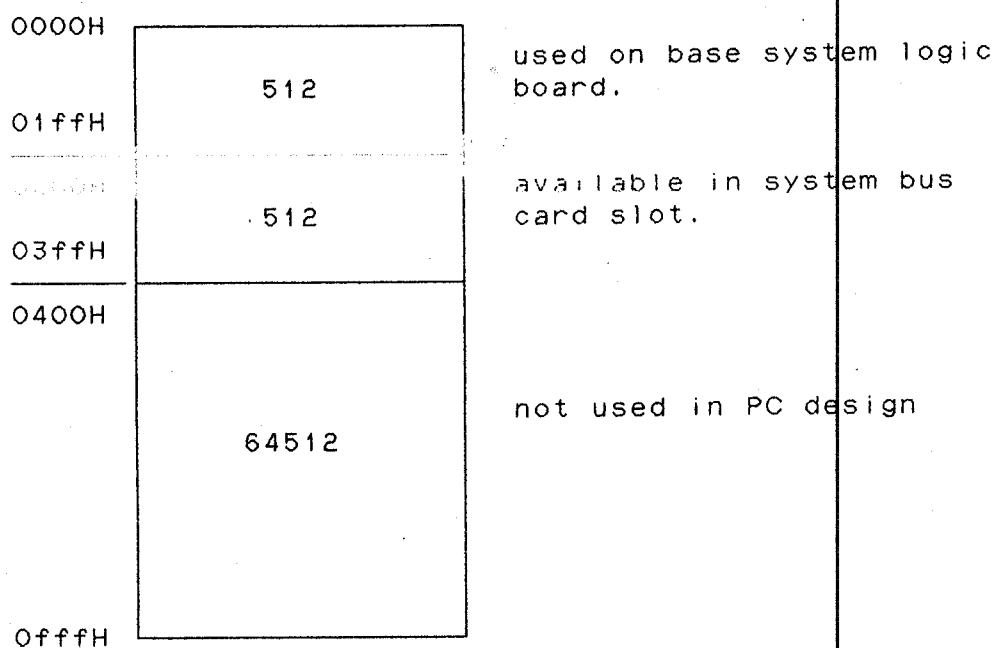
- Power Supply dan Ground

Ada 4 macam tegangan supply yang dapat digunakan oleh peralatan eksternal yang terpasang pada slot, yaitu +5 volt, -5 volt, +12 volt dan -12 volt dengan daya maksimum 130 watt.

### 11.5.2 PENGALAMATAN I/O PORT PADA IBM PC

Setiap peralatan I/O mempunyai alamat yang unik, yang disebut alamat I/O port. CPU 8088 yang terdapat pada sistem board IBM PC dapat mengakses peralatan I/O pada satu saat dengan memberikan alamat I/O port yang sesuai melalui jalur alamat.

CPU 8088 dirancang untuk dapat mengakses 65.536 alamat I/O port. Tetapi pada IBM PC, alamat yang digunakan untuk I/O port hanya 10 bit terendah saja yaitu A0 sampai A9. Dengan begitu hanya 1024 alamat I/O port saja yang dapat diakses. Pemakaian 1024 alamat I/O port ini dibagi menjadi dua bagian yaitu 512 lokasi digunakan oleh sistem board dan 512 yang lain digunakan oleh expansion slot atau diluar sistem board.



Gambar 2.14<sup>24)</sup>

#### Pembagian alamat I/O Port pada IBM PC

Untuk menentukan apakah I/O port yang dipilih berada pada sistem board atau diluar sistem board, digunakan bit tertinggi dari 10 bit alamat I/O port

<sup>24)</sup> ibid hal. 127

yang disediakan yaitu A9. Bila A9 sama dengan logic high ('1') maka yang dipilih adalah I/O port diluar sistem board. Dengan demikian alamat 0000h - 01FFh digunakan untuk keperluan sistem board. Sedangkan alamat 0200h - 03FFh untuk I/O port diluar sistem board. Pembagian ini dapat digambarkan seperti ditunjukkan pada gambar 2.14

TABEL 2.2<sup>25)</sup>

## I/O PORT DALAM SISTEM BOARD

ALAMAT(HEX)	KETERANGAN
0000 - 000F	DMA chip 8237
0010 - 001F	Belum digunakan
0020 - 0021	Interrupt chip 8259
0022 - 003F	Belum digunakan
0040 - 0043	Timer chip 8253
0044 - 005F	Belum digunakan
0060 - 0063	PPI chip 8255
0064 - 007F	Belum digunakan
0080 - 0083	DMA page register
0084 - 009F	Belum digunakan
00A0 - 00A0	NMI mask bit
00A1 - 01FF	Belum digunakan

Dari 1024 address I/O port yang disediakan, tidak seluruhnya dipakai. Pembagian lokasi untuk I/O port dalam sistem board ditunjukkan pada tabel 2.2. Sedang pembagian lokasi untuk I/O port diluar sistem board ditunjukkan pada tabel 2.3

Dalam perencanaan pembuatan interface yang akan dipakai pada unit I/O, akan dipilih lokasi I/O diluar

---

<sup>25)</sup> ibid hal.128

TABEL 2.3<sup>26)</sup>

## I/O PORT DI LUAR SISTEM BOARD

ALAMAT (HEX)	KETERANGAN
0200 - 020F	Game kontroller
0210 - 0217	Unit ekspansi
0218 - 02F7	Belum digunakan
0278 - 02FF	Asyn. comm #2
0300 - 031F	Belum digunakan
0320 - 032F	Fixed disk
0330 - 0377	Belum digunakan
0378 - 037F	Paralel port
0380 - 038F	SLDC COMM
0390 - 03AF	Belum digunakan
03B0 - 03BF	Monochrome/printer
03C0 - 03CF	Belum digunakan
03D0 - 03DF	Color graphics
03E0 - 03EF	Belum digunakan
03F0 - 03F7	Disk drive
03F8 - 03FF	Belum digunakan

sistem board yang belum digunakan. Pemilihan alamat didasarkan pada kemudahan dalam rangkaian pengkode sedemikian sehingga pemakaian komponen logika dapat ditekan seminimal mungkin.

Untuk keperluan pemindahan data dari mikrokomputer ke-memori RAM ini selain rangkaian pengkode dan penyangga juga dibutuhkan rangkaian latch yang akan menahan data yang dikeluarkan oleh IBM PC pada saat alamat I/O port dipilih. Jalur output dari latch tersebut dihubungkan dengan jalur alamat dari memori RAM. Pada saat data pertama yang dipilih oleh sistem I/O port dikeluarkan oleh IBM PC, maka data

---

<sup>26)</sup> ibid hal. 129

tersebut akan ditahan dan di-output-kan ke jalur alamat dari memori RAM, dan data kedua yang dikeluarkan oleh IBM PC tersebut akan dihubungkan langsung ke jalur data dari memori RAM. Proses ini berlangsung terus sampai semua data model selesai dipindahkan ke RAM.

## 11.6 OPTOCOUPLER

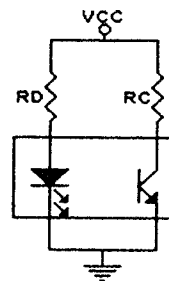
Optocoupler termasuk jenis semikonduktor yang didalamnya terdapat satu sumber cahaya berupa LED infra merah, dan sebuah detektor cahaya berupa transistor.

Sesuai dengan fungsinya yang sering dipakai sebagai isolator elektronik, maka optocoupler dikenal dengan sebutan opto-isolator.

Optocoupler terdiri dari *light emitter*, *light path* dan detektor cahaya (photo transistor) tergabung menjadi satu komponen yang tertutup. Struktur dari optocoupler seperti yang terlihat pada gambar 2.15 hanya bisa memungkinkan transfer satu jalur dari sinyal elektrik dari salah satu jalur port.

Prinsip kerja dari optocoupler dapat dijelaskan sebagai berikut : Sinyal yang masuk pada pin input optocoupler diubah kedalam bentuk cahaya oleh LED. Kuat lemahnya cahaya yang dihasilkan oleh LED tergantung dengan besar sinyal input yang diterima oleh LED. Pancaran cahaya yang dihasilkan oleh LED akan mengenai foto-transistor yang ada didepannya, dan kemudian oleh foto-transistor cahaya dari LED ini diubah menjadi

sinyal listrik. Sinyal listrik yang dihasilkan oleh foto-transistor merupakan output dari optocoupler.



Gambar 2.15

Simbol optocoupler

Kelebihan-kelebihan pemakaian optocoupler antara lain:

1. adanya isolasi antara rangkaian dalam dan luar.
2. switch dengan kecepatan tinggi.
3. kekuatan terhadap guncangan atau getaran mekanik cukup tinggi.
4. ukurannya kecil dan bentuknya sederhana sehingga mempermudah dalam pemakaian dan efisien.

## 11.7 RELAY

Motor dc yang digunakan dalam tugas akhir ini mempunyai arus kerja yang besar. Disamping itu frekuensi putaran dari motor tersebut sering mempengaruhi kerja dari bagian kontroller. Sehingga untuk mengatasi masalah tersebut dibutuhkan suatu isolasi yang memisahkan antara bagian kontroller dan

bagian penggerak motor.

Relay yang bekerja berdasarkan ada dan tidaknya medan magnet mampu mengatasi masalah tersebut. Keuntungan dari penggunaan relay adalah dari kesederhaan rangkaian pengendalinya.

### Relay Histerisis<sup>27)</sup>

Untuk mengaktifkan relay, arus kumparan harus mencapai suatu harga yang disebut *pull in current*, untuk menggerakkan armature dan men'switch' kontak. Dan untuk kembali pada keadaan semula arus kumparan harus turun sampai pada suatu harga terendah yang menyebabkan armature kembali pada keadaan semula. Harga arus terendah ini disebut *drop out current*. Kejadian ini digambarkan pada gambar 2.16

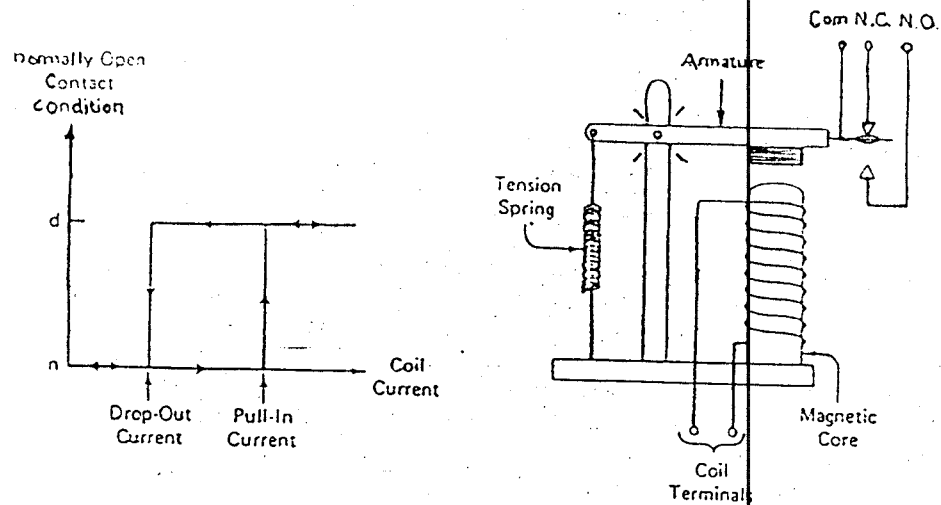
Alasan adanya perbedaan antara arus pull in dan drop out dapat dimengerti dengan melihat gambar dari relay tersebut yang diperlihatkan pada gambar diatas. Per menyebabkan armature naik menjauhi inti besi ketika relay tidak diaktifkan. Ini akan menimbulkan jarak antara inti besi dengan armature. Ketika arus kumparan mulai mengalir, harus dihasilkan medan magnet yang cukup besar untuk menarik armature kebawah. Dan ini menghadapi 2 masalah, yaitu :

1. Adanya jarak antara loop medan magnet. Ini menghasilkan medan magnet yang lebih lemah dari pada

---

<sup>27)</sup> Timothy J. Maloney. Industrial Solid State Electronics., Prentice Hall. 1979 hal. 348





Gambar 2.16

a. relay histerisis b. bagian dari relay

loop medan magnetik yang tanpa jarak.

2. Lemahnya gaya tarik antara inti dengan armature yang disebabkan adanya jarak antar kutub. Apabila kutub magnetik tersebut lebih jauh maka gaya tarik juga akan lebih lemah.

Jika arus kumparan cukup besar maka akan dihasilkan medan magnet yang cukup besar untuk mengatasi masalah tersebut.

#### 11.8 MOTOR ARUS SEARAH<sup>28)</sup>

Menurut hubungan kumparan medannya maka motor arus searah diklasifikasikan dalam 4 tipe, sebagai

<sup>28)</sup> ibid hal.595

berikut :

- motor dc penguatan terpisah
- motor dc shunt
- motor dc seri
- motor dc kompon yang mempunyai medan shunt dan medan seri.

#### **Motor DC Penguatan Terpisah**

Mempunyai kumparan medan yang disuply oleh sumber lain yang bebas dan tidak tergantung beban atau tegangan drop didalam jangkar. Kecepatan praktis tetap pada seluruh range.

#### **Motor DC Shunt**

Mempunyai kumparan medan yang dihubungkan paralel dengan jangkar motor itu sendiri. Kenaikan beban akan menyebabkan drop torsi, drop kecepatan kecil (Kecepatan relatif tetap).

#### **Motor DC Seri**

Mempunyai kumparan medan yang dihubung seri dengan kumparan jangkar, dimana kenaikan beban akan menyebabkan drop torsi, drop kecepatan yang besar, tidak dapat berputar pada beban ringan/ tanpa beban sebab akan dapat terjadi kerusakan karena dengan kecepatan yang sangat tinggi akan menyebabkan gaya sentrifugal yang besar yang dapat merusak mekanisme sehingga beban harus dikopel langsung dengan motor dc seri.

### Motor Kompon

Mempunyai kumparan medan seri dan shunt, dimana karakteristiknya ditentukan oleh kekuatan relatif dari kedua medan jadi terletak antara karakteristik motor dc shunt dan motor dc seri.

Dari karakteristik-karakteristik tersebut diatas, maka dalam tugas akhir ini digunakan motor dc shunt sebagai penggerak tool, dikarenakan kecepatannya yang relatif konstan dengan adanya perubahan beban.

#### 11.8.1 PENGOPERASIAN DAN KARAKTERISTIK MOTOR DC<sup>29</sup>

Motor dc sangat penting dalam kontrol industri disebabkan lebih mudah diatur kecepatannya dibandingkan motor ac. Gambar 2.17 menunjukkan simbol dari motor dc shunt. Medan kumparan motor digambarkan sebagai koil, yang secara fisik, terdiri dari banyak belitan kawat tipis diseputar kutub medan. Kutub medan terbuat dari inti besi ferromagnet, yang berfungsi sebagai stator.

Hambatan yang tinggi dari kumparan medan membatasi arus medan supaya mempunyai harga yang cukup kecil, sehingga mengijinkan kumparan medan dihubungkan langsung kejalur dc supply.

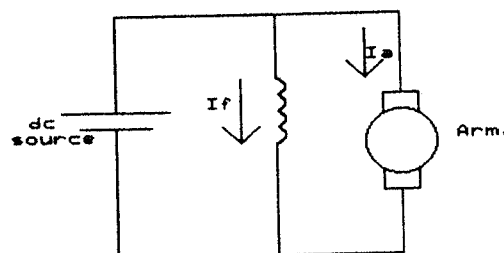
---

<sup>29)</sup> ibid hal. 596

Kumparan medan tidak terpengaruh oleh perubahan keadaan dari armature. Sehingga, ketika arus dari armature berubah dikarenakan perubahan dari besarnya beban, arus kumparan medan tetap konstan dan kekuatan medan magnet juga tetap konstan. Arus medan dapat dihitung dengan mudah menggunakan hukum ohm sebagai

$$I_f = \frac{V_s}{R_f} \quad \dots 2.1$$

dimana,  $V_s$  adalah tegangan sumber yang digunakan di kumparan medan dan  $R_f$  adalah hambatan dc dari kumparan.



Gambar 2.17

Simbol motor dc

Kumparan armature dari dc motor mempunyai belitan yang relatif lebih sedikit, terdiri dari kawat yang lebih tipis, sehingga mempunyai hambatan dc yang kecil, biasanya sekitar  $1 \Omega$ . Ketika power pertama kali diberikan pada armature, hanya hambatan dc ohmic yang membatasi arus, sehingga arus cukup besar. Bagaimanapun

Juga ketika motor mulai menaikkan kecepatan, motor mulai menginduksikan EMF lawan. EMF lawan ini melawan tegangan sumber dan membatasi arus armature. Ketika dc motor ini telah mencapai kecepatan normal, EMF lawannya sekitar 90%, sama dengan tegangan yang dipakai pada armature.

Tegangan drop  $I_R$  yang melalui hambatan kumparan armature sekitar 10% dari tegangan sumber, dengan mengabaikan tegangan drop pada sikat karbon.

Besarnya EMF lawan yang dihasilkan oleh kumparan armature tergantung dari dua hal, yaitu :

- . Kuat dari medan magnet. Makin besar medan magnet, makin besar EMF lawan yang dihasilkan.
- . Kecepatan putaran. Semakin besar kecepatan, semakin besar EMF lawan yang dihasilkan.

Hukum tegangan kirchoff untuk loop armature ditunjukkan pada persamaan 2.2 yang mana menyatakan bahwa tegangan yang digunakan pada armature sebanding dengan jumlah tegangan drop pada armature. Jumlah tegangan drop pada kumparan armature sebanding dengan EMF lawan ditambah dengan tegangan drop  $I_R$ , dengan mengabaikan tegangan drop pada sikat.

$$V_s = E_c + I_a R_a$$

...2.2

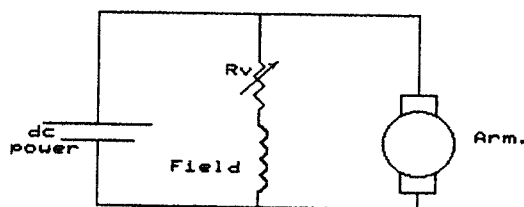
dimana  $I_a$  adalah arus armature dan  $R_a$  adalah hambatan dari armature, sedang  $E_c$  adalah EMF lawan.

### 11.8.2 PENGATURAN KECEPATAN DARI MOTOR DC SHUNT<sup>30)</sup>

Secara dasar, ada dua cara untuk mengontrol kecepatan dari motor dc shunt :

1. Pengaturan tegangan dan arus yang diberikan pada kumparan medan.

Saat tegangan medan bertambah besar, motor menurun kecepatannya. metode ini ditunjukkan pada gambar 2.18



Gambar 2.18

Pengaturan tegangan dan arus pada kumparan medan

Metode ini mempunyai kebaikan, karena untuk pengaturan tegangan dapat digunakan rheostat yang kecil dan murah, sebab arus pada kumparan medan kecil sehingga rheostat tidak menyerap energi yang besar.

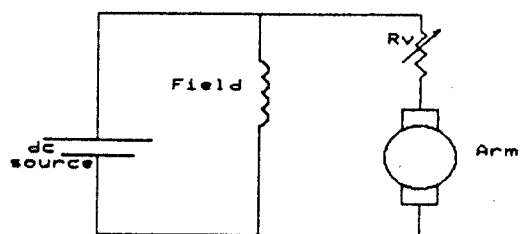
<sup>30)</sup> ibid hal.598

Bagaimanapun juga terdapat kerugian pada pengontrolan tegangan ini. Untuk memperbesar kecepatan dilakukan dengan memperkecil  $I_f$  yang akan menyebabkan lemahnya medan magnet dan hal ini akan memperkecil torsi yang dihasilkan oleh motor. .lmi

2. Pengaturan tegangan dan arus yang diberikan pada armature.

Ketika tegangan armature bertambah besar, motor bertambah kecepatannya. Metode ini ditunjukkan pada gambar 2.19.

Masalah yang timbul pada metode ini adalah besar dan mahalnya rheostat, dikarenakan rheostat ini harus menangani arus armature yang cukup besar, sehinggan ini akan memboroskan energi.



Gambar 2.19

Pengaturan tegangan dan arus pada armature

## BAB III PERENCANAAN

---

### III.1 PENDAHULUAN

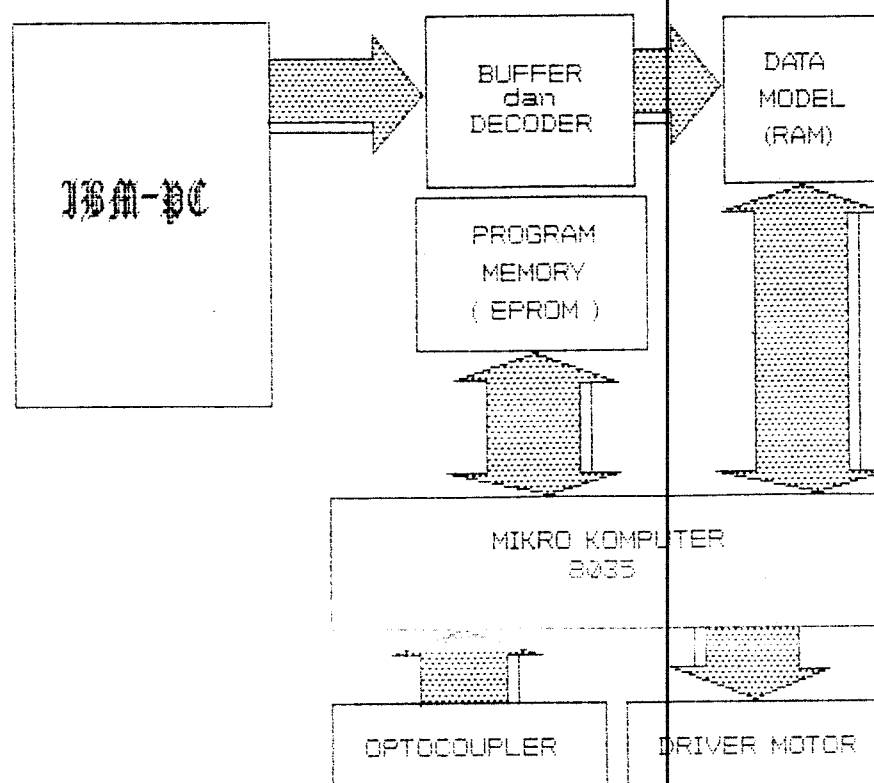
Pada bab perencanaan ini akan dijelaskan tentang perencanaan yang berhubungan dengan pembuatan alat. Komponen-komponen yang digunakan secara umum telah dijelaskan dalam bab II dan secara terperinci akan dibahas pada bab ini. Software yang digunakan untuk mengontrol seluruh kerja sistem akan dijelaskan pada akhir bab yang berupa flowchart, sedang listing program dari sistem diberikan dalam lampiran.

### III.2 BLOK DIAGRAM SISTEM

Blok diagram dari perencanaan alat ini terdiri dari beberapa bagian yang semuanya ditunjukkan dalam gambar 3.1. Blok diagram tersebut terdiri dari bagian IBM PC yang merupakan sarana untuk membuat data model, card interface yang merupakan bagian penghubung antara komputer IBM PC dengan peralatan luar, bagian RAM yang merupakan sarana penyimpan data model, bagian EPROM yang merupakan sarana penyimpan program dari mikrokomputer, bagian mikrokomputer yang merupakan



pusat pengontrol utama dari sistem dan yang terakhir bagian sensor dan bagian driver dari motor dc. Sub bab berikut akan menerangkan fungsi dari masing-masing blok.



Gambar 3.1

Blok diagram sistem

### III.3 PRINSIP KERJA SISTEM

Cara kerja dari peralatan yang dibuat ini meliputi urutan kerja sebagai berikut:

- . Membuat model didalam IBM PC dengan input sumbu X dan sumbu y. Pembuatan model dilakukan dalam IBM PC dikarenakan kemudahannya dalam operasi aritmatika seperti pengurangan, penjumlahan, perkalian dan kwadrat maupun akar. Yang mana operasi ini sangat dibutuhkan dalam pembuatan model.
- . Proses penyimpanan data model dimemori adalah sebagai berikut. Data pertama yang dikeluarkan oleh IBM PC akan di-input-kan ke jalur alamat dari RAM melalui rangkaian latch dan kemudian data kedua yang dikeluarkan oleh IBM PC akan di-input-kan langsung ke jalur alamat dari RAM. Proses ini berlangsung terus sampai semua data model yang telah dibuat, disimpan kedalam memori RAM. Data yang telah disimpan di RAM ini kemudian diolah oleh kontroller untuk menggerakkan motor yang mengatur posisi tool.
- . Pertama sensor home posisi Y akan mendeteksi apakah posisi Y dari tool sudah sampai pada awal, jika belum maka motor posisi Y akan menggerakkan tool sampai pada posisi awal. Setelah posisi awal Y tercapai, maka kontroller akan meneruskan untuk mendeteksi posisi awal dari posisi X. Setelah proses posisi awal selesai, maka kontroller akan meneruskan pada tahap pendeteksian ketebalan bahan kerja. Proses ini diperlukan supaya

kontroller tidak melakukan kerja yang mubadzir, dimana mesin melakukan pemakanan tetapi tidak ada bahan kerja yang dimakan.

- Kontroller akan mendeteksi penekanan switch adjusting (pengaturan). Adjusting ini akan menentukan majunya posisi tool setiap proses pemakanan, yang biasanya besarnya ditentukan oleh jenis bahan yang akan dibubut dan jenis tool yang dipakai. Pada perencanaan ini diberikan 3 pilihan untuk adjusting, yaitu untuk kemajuan tool setiap 4mm, 3mm dan 2mm.

- Setelah kontroller selesai mendeteksi proses penekanan, maka proses pembubutan dimulai. Dimana kontroller akan mengambil data model yang telah dibuat sebelumnya, dan kemudian data tersebut dikurangkan dengan ketebalan bahan dan bandingkan dengan adjusting. Jika data hasil lebih besar dari adjusting maka data baru sama dengan adjusting, dan jika tidak maka data baru sama dengan data hasil. Simpan data baru tersebut kealamat atas dari RAM (80h). Proses berulang terus sampai semua data selesai diolah.

Posisi Y dari tool akan maju sebesar adjusting dan counter akan diisi sesuai dengan besar adjusting. Counter akan dibandingkan dengan data baru, jika counter tidak sama dengan data baru, maka motor

posisi Y akan menggerakkan tool sampai didapatkan keadaan dimana counter sama dengan data baru. Setelah sama maka kontroller akan menghentikan motor posisi Y dan kontroller akan menggerakkan motor posisi X untuk melakukan proses pemakanan dan berhenti setelah melakukan proses pemakanan sebesar 2mm. Setelah proses pemakanan selesai maka kontroller akan mengambil data baru lagi dan dibandingkan dengan counter dan proses berulang sampai dicapai data model maksimum. Setelah semua data diambil, maka motor pengatur posisi feeding akan bergerak keposisi awal begitu juga dengan motor pengatur posisi tool. Setelah tool kembali ke posisi awal, maka kontroller akan mengurangi kembali data hasil yang disimpan di-RAM dengan data adjusting dan proses awal terulang kembali sampai data adjusting lebih besar dari data hasil terakhir yang disimpan di-RAM.

### III.3.1 BAGIAN SISTEM KONTROLLER

#### a. Mikrokontroller 8035

Komponen ini digunakan sebagai pusat pengontrol seluruh kerja dari sistem, meliputi pengambilan data dari RAM, pengolahan data tersebut, menggerakkan motor dan lain-lain.

b. EPROM 2716

Adalah komponen memori dari jenis non-volatile (tak dapat dibersihkan) yang berisi program untuk mengontrol kerja sistem.

c. RAM 6116

Adalah jenis memori volatile (dapat dibersihkan) yang berfungsi untuk menyimpan data yang diambil dari IBM PC dan kemudian data tersebut diolah oleh kotroller untuk menggerakkan motor.

### III.3.2 BAGIAN INTERFACE UNTUK IBM PC-XT

a. Pengkode 74LS138

Digunakan untuk memilih alamat dari I/O port. Dalam tugas akhir ini dipilih alamat 0330h - 0333h untuk mengaktifkan perangkat luar dalam hal ini RAM dan komponen latch 74LS377.

b. Penyangga 74LS245

Komponen ini digunakan untuk mengisolasi sistem komputer IBM PC dengan peralatan luar, pada saat peralatan luar tidak diaktifkan.

c. Latch 74LS377

Komponen ini digunakan untuk menahan data yang dikeluarkan oleh IBM PC pada saat alamat I/O port dipilih, dan data tersebut akan di-output-kan ke jalur alamat dari RAM.

### III.3.3 BAGIAN DRIVER DAN SENSOR

#### a. Transistor

Transistor digunakan sebagai switch elektronik yang akan mengaktifkan relay.

#### b. Opto coupler

Optocoupler digunakan sebagai sensor jarak. Sebagai sensor jarak optocoupler dihubungkan ke bagian counter yang terdapat dalam chip 8035.

Selain itu optocoupler ini digunakan untuk mengetahui posisi awal (home posisi) dari peralatan.

### III.4 PERANGKAT KERAS

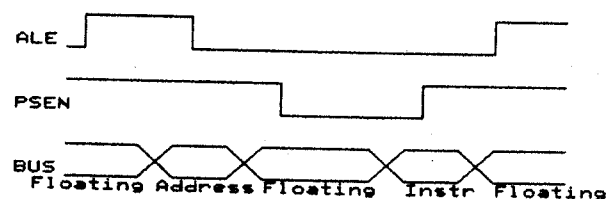
#### III.4.1 PERENCANAAN SISTEM KONTROLLER

Sebagai kontroller digunakan mikrokomputer 8035 yang mempunyai spesifikasi sebagai berikut :

- menggunakan supply tegangan tunggal 5V
- terdapat RAM internal sebesar 64 x 8 bit
- fasilitas untuk input output sebanyak 27 jalur yang terdiri dari 2 x 8 bit port, 1 x 8 bit bus, 1 x 8 bit timer, 1 x 8 bit counter dan jalur interrupt.
- menggunakan program memori luar sebesar 4K x 8 bit
- dapat ditambahkan data memori luar sebesar 256 x 8 bit.

### III.4.2 PERLUASAN PROGRAM MEMORI

Disebabkan mikrokomputer type 8035 ini tidak mempunyai internal program memori, sehingga untuk keperluan tersebut digunakan memori luar. Seperti ditunjukkan pada gambar 3.2, urutan proses pengambilan instruksi dari program memori luar adalah sebagai berikut :<sup>30)</sup>



Gambar 3.2

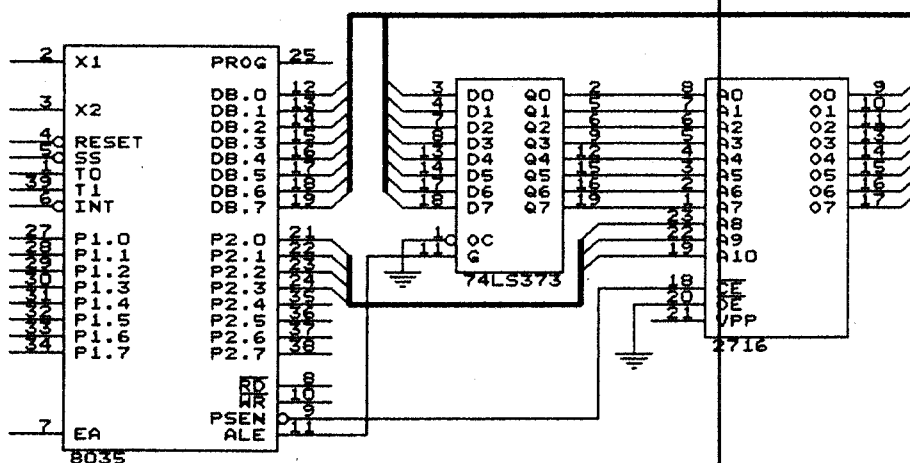
Pengambilan instruksi dari program memori luar

1. Isi dari 12 bit program counter akan dikeluarkan ke BUS dan port 2 bagian bawah (P20 - P23).
2. *Address Latch Enable* (ALE) akan menunjukkan saat alamat valid.
3. *Program Send Enable* (PSEN) menunjukkan bahwa instruksi luar sedang diambil dan digunakan untuk mengaktifkan peralatan memori luar.
4. BUS berubah menjadi mode input dan prosessor akan

<sup>30)</sup>...., Single Chip 8-Bit Microcontroller., Philips Electronic Componen Materials., 1986. Hal. 3-1

menerima data instruksi 8 bit.

Untuk keperluan ini digunakan EPROM type 2716 yang merupakan memori bersifat dapat dihapus dan diprogram. Jalur data dan alamat bawah pada mikrokomputer 8035 ini tergabung menjadi satu yang disebut Bus Address-Data sehingga untuk memisahkannya digunakan rangkaian latch, dan pada perencanaan ini digunakan komponen latch type 74LS373 dengan input enablenya dihubungkan ke pin ALE dari mikrokomputer 8035. Untuk alamat atas dari EPROM A8 - A11 dihubungkan langsung dengan port P20-P23 dari mikrokomputer tersebut. Pin EA (External Access) dihubungkan ke Vcc. Perencanaan sistem program memory ini dapat dilihat pada gambar 3.3



Gambar 3.3

Rangkaian program memory luar

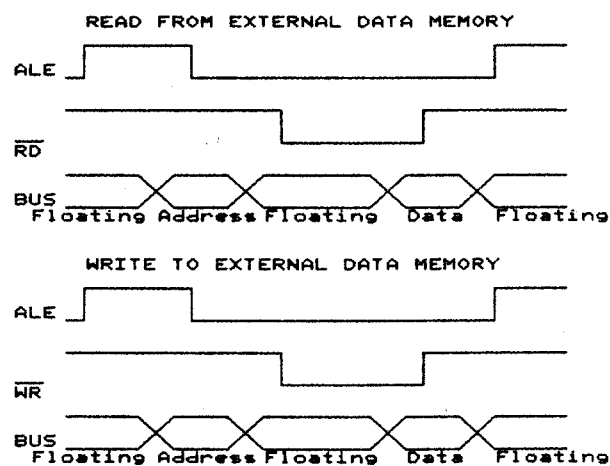


### III.4.3 PERLUASAN DATA MEMORI

Data memori pada 8035 dapat diperluas sampai dengan 256 byte, dimana alamat dan data tersebut dipindahkan melalui jalur BUS.

#### Cycle Penulisan Dan Pembacaan Data

Semua alamat dan data dipindahkan melalui 8 jalur BUS. Gambar 3.4 memperlihatkan cycle dari pembacaan dan penulisan data. Proses pembacaan dan penulisan data tersebut adalah sebagai berikut :<sup>31)</sup>



Gambar 3.4

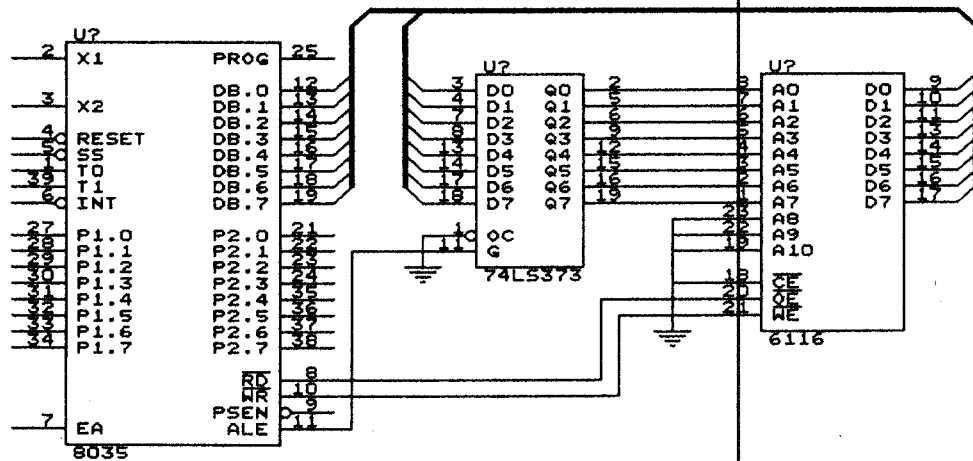
Timing data memory luar

1. Isi dari register R0 dan R1 dikeluarkan ke BUS.
2. *Address Latch Enable* (ALE) menunjukkan validnya alamat.

<sup>31)</sup> ibid hal.3-4

3. Pulsa RD atau WR dari pin 8035 akan menunjukkan bahwa data siap dikirim atau diterima. Output data valid pada akhir dari kenaikan pulsa WR dan input data harus valid pada saat akhir dari kenaikan pulsa RD.
4. Data (8 bit) dipindahkan kedalam atau keluar dari BUS.

#### Pengalamatan Data Memori Luar



Gambar 3.5

#### Rangkaian sistem data memori luar

Data memori luar diakses dengan menggunakan 2 cycle instruksi MOVX A, @R dan MOVX @R, A yang akan memindahkan 8 bit data antara akumulator dan lokasi memori luar yang alamatnya ditunjukkan oleh isi salah satu dari RAM Pointer Register, R0 atau R1. Untuk perluasan data memori digunakan RAM tipe 6116

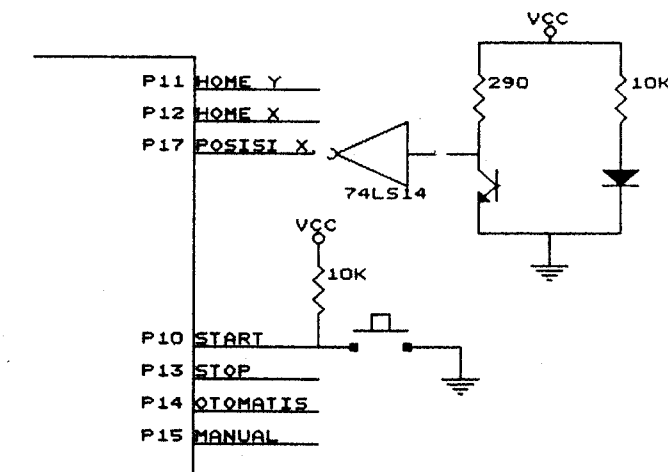
yang mempunyai kapasitas memori sebesar 2Kb. Gambaran perluasan data memori diperlihatkan pada gambar 3.5

#### III.4.4 INPUT - OUTPUT PORT

Mikrokontroller 8035 mempunyai 27 jalur yang dapat digunakan sebagai input atau output. Jalur jalur ini dikelompokkan dalam 3 port yang masing-masing terdiri dari 8 jalur yang dapat berlaku sebagai input, output maupun bidirectional port dan 3 test input yang terdiri dari INT, T0 (timer) dan T1 (counter).

##### Port 1 dan Port 2

Pada perencanaan ini port 1 digunakan sebagai input port. Port ini dihubungkan dengan output dari optocoupler dan switch-switch. Sedang port 2 digunakan sebagai output port, yang akan dihubungkan ke transistor yang digunakan sebagai switch untuk mengaktifkan relay. Proses pengambilan data input dilakukan dengan memindahkan isi port 1 ke akumulator dengan instruksi IN A, P1. sedang proses pengambilan data output dilakukan dengan memindahkan isi akumulator ke port 2 dengan menggunakan instruksi OUTL P2, A. Gambar 3.6 menunjukkan perencanaan port 1 sebagai input port, sedang gambaran mengenai penggunaan port 2 sebagai output port (penggerak motor) ditunjukkan pada bahasan transistor sebagai pengemudi relay.



Gambar 3.6

Port 1 sebagai input port

### INT, T0 dan T1

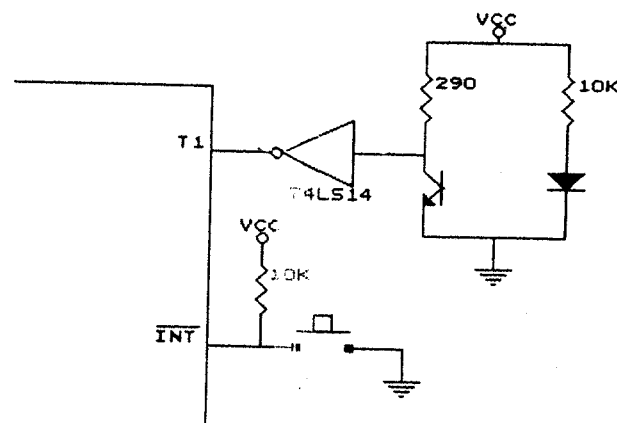
Interrupt digunakan apabila terjadi gangguan pada mesin, sehingga dilakukan penghentian mendadak, sebelum mesin selesai melakukan pekerjaannya. Pada saat di-interrupt maka mikrokomputer akan menyimpan register kerja (R0 - R7) dengan menggunakan instruksi SEL RB1. Instruksi ini akan menyimpan register kerja R0 - R7 dan untuk proses selanjutnya digunakan register alternatif (R0' - R7') yang terletak internal RAM lokasi 24 - 31.

T0 yang merupakan fasilitas timer yang disediakan oleh mikrokomputer 8035 dipergunakan sebagai delay. Variasi delay antara 60 mikrodetik sampai 15 milidetik (256 hitungan) dapat dihasilkan dengan menggunakan kristal clock 8 MHz. Delay ini digunakan untuk menghilangkan debounce yang dihasilkan oleh

switch input, disamping itu juga digunakan untuk memberi waktu motor sebelum arah putarannya dibalik.

T1 merupakan fasilitas counter yang disediakan oleh mikrokomputer. Counter disini akan dihubungkan dengan optocoupler untuk mengetahui posisi dari tool. Perubahan logika dari high ke low pada input T1 akan menyebabkan kenaikan isi dari counter.

Gambar 3.7 menunjukkan gambaran perencanaan dari interrupt dan counter.



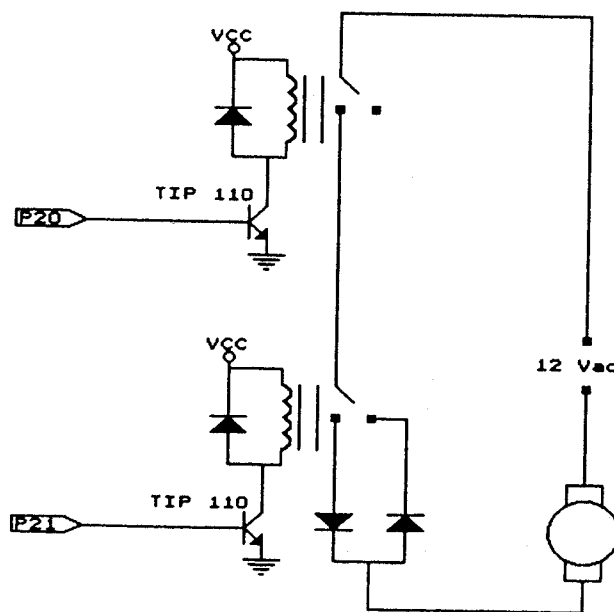
Gambar 3.7

Interrupt dan counter

### III.5 TRANSISTOR SEBAGAI PENGEMUDI RELAY

Untuk mengaktifkan dan mengatur arah putaran dari motor dc yang akan menggerakkan tool dan yang melakukan gerakan pemakanan digunakan relay. Keuntungan dari penggunaan relay ini adalah untuk mengisolasi

antara supply yang digunakan untuk menggerakkan motor dan supply yang digunakan untuk mengaktifkan bagian controller, disamping itu penggunaan relay ini lebih ekonomis dan lebih mudah dibandingkan dengan komponen elektronik yang lain, seperti TRIAC atau SCR yang dipekerjakan sebagai switch.



Gambar 3.8

Transistor sebagai penggerak relay

Relay yang digunakan disini adalah relay dc yang mempunyai tegangan kerja 12 volt dan arus kerja 400 mA. Untuk menggerakkan relay ini digunakan transistor yang dipekerjakan sebagai switch. Pada perencanaan ini, transistor yang digunakan tipe TIP 110, yang merupakan jenis darlington. Gambar 3.8

menunjukkan rangkaian penggerak relay yang direncanakan.

Berikut ini adalah cara kerja dari rangkaian tersebut. Pin output port memberikan arus ke basis dari transistor. Arus basis ini menghasilkan arus kolektor sebesar arus basis dikalikan dengan beta dari Q1. Arus kolektor dari Q1 menjadi arus basis dari Q2 dan diperkuat dengan dengan penguatan arus dari Q2.

Hasil dari keseluruhan rangkaian ini seperti transistor tunggal yang mempunyai penguatan arus sebesar beta Q1 dikalikan beta Q2 dengan tegangan basis emitor sebesar 1.4 volt. TIP 110 mempunyai beta minimum sebesar 1000 pada IC sebesar 1 A. Jika dibutuhkan arus sebesar 200 mA untuk menggerakkan relay maka arus minimum yang harus diberikan oleh output port sebesar  $200 \text{ mA} / 1000$  atau sebesar 0.2 mA dan ini cukup mudah dilakukan oleh port dari 8035 tersebut.

#### III.6.1 INTERFACING PENYIMPANAN MODEL DI RAM

Mesin bubut yang direncanakan bekerja berdasarkan model yang dibuat sebelumnya. Disebabkan untuk pembuatan model tersebut dibutuhkan banyak operasi matematik seperti operasi perkalian, mencari hasil akar dan sebagainya. Sehingga untuk memudahkan, pengolahan data model tersebut dilakukan di IBM PC

Kemudian data model tersebut ditransferkan ke-RAM.

Untuk menghubungkan IBM PC tersebut dengan peralatan luar dalam hal ini RAM maka dibutuhkan rangkaian tambahan, yaitu rangkaian pengkode yang akan memilih alamat-alamat yang tidak digunakan oleh IBM PC dan disediakan untuk sistem unit I/O dan disamping itu juga dibutuhkan peralatan input output yang akan dihubungkan ke alamat, data dan kontrol dari RAM.

#### III.6.2 I/O Mapping Pada IBM PC

I/O mapping diperlukan untuk memberikan alamat pada komponen penunjang dan peralatan input/output. Hal ini dilakukan agar IBM PC dapat memperlakukan peralatan I/O seperti pada peralatan memori yaitu dapat dibaca dan ditulis dengan menunjukkan pada alamat tertentu.

Alamat yang digunakan dalam alat ini yaitu alamat dari lokasi memori IBM PC yang tersedia untuk unit I/O slot IBM PC, seperti tampak pada gambar 3.9. Dari tabel ini dapat dilihat bahwa lokasi dengan alamat 0200h s/d 03FFh khusus disediakan untuk unit interface yang ditancapkan pada slot IBM PC.

Dari gambar 3.9 yang memperlihatkan alamat I/O port pada slot, dipilih lokasi dengan alamat 0330h s/d 0333h.



ALAMAT (HEX)	KETERANGAN
0200 - 020F	Game kontroller
0210 - 021F	Unit ekspansi
0218 - 02F7	Belum digunakan
0278 - 02FF	Asyn. comm #2
0300 - 031F	Belum digunakan
0320 - 032F	Fixed disk
0330 - 0377	Belum digunakan
0378 - 037F	Paralel port
0380 - 038F	SLDC COMM
0390 - 03AF	Belum digunakan
03B0 - 03BF	Monochrome/printer
03C0 - 03CF	Belum digunakan
03D0 - 03DF	Color graphics
03E0 - 03EF	Belum digunakan
03F0 - 03F7	Disk drive
03F8 - 03FF	Belum digunakan

Gambar 3.9

Alamat I/O port pada slot

### III.6.3 SISTEM PENGKODEAN

Dalam perencanaan ini digunakan dekoder type 74LS138 untuk membentuk alamat dan peralatan input output pada interface. Tabel kebenaran dari dekoder ini diperlihatkan pada tabel 3.1. Dari tabel ini dapat dilihat bahwa dekoder tersebut mempunyai 3 jalur input (A, B, C), 3 jalur enable dan 8 jalur output, dan dengan kombinasi dari 3 jalur input dapat dipilih salah satu jalur output yang aktif.

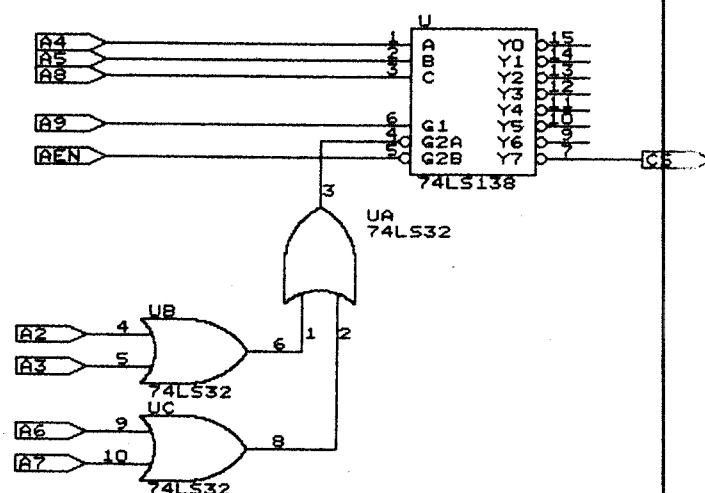
Gambar 3.10 memperlihatkan rangkaian dari sistem pengkode tersebut. Pada gambar rangkaian diatas output Y7 aktif bila alamat 0330 - 0333 dipilih.

Penambahan AEN pada pendekodean alamat ini, dilakukan untuk menghindari terjadinya proses pendekodean bersama dengan proses DMA, karena hal ini bisa menyebabkan kesalahan.

TABEL 3.1

TABEL KEBENARAN DECODER 74LS138

Input					Output							
Enable		Select										
G1	G2	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	H	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L



Gambar 3.10

Rangkaian sistem pengkodean

#### III.6.4 PENYANGGA

Penyangga dibutuhkan untuk memperkuat sinyal pada jalur data. Penyangga yang digunakan disini yaitu Penyangga dua arah dengan type 74LS245, dikarenakan pada jalur data ini terjadi perpindahan data output (dari IBM PC) dan data input (dari interface).

TABEL 3.2

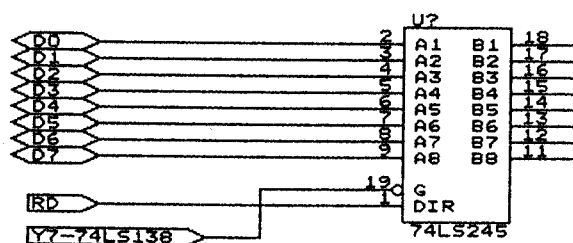
TABEL KEBENARAN 74LS245

G	DIR	
L	H	A —————> B
L	L	B <———— A
H	H	TRISTATE
H	L	TRISTATE

Tabel Kebenaran dari buffer ini diperlihatkan pada tabel 3.2. Dari tabel tersebut dapat dilihat, bahwa untuk mengaktifkan buffer tersebut dengan memberi sinyal low pada enable (G), dan DIR menentukan arah dari perpindahan data. Jika DIR diberi low maka data akan berpindah dari sisi B ke sisi A, dan begitu juga sebaliknya.

Gambar dibawah menunjukkan cara menghubungkan pin-pin 74LS245. A0 s/d A7 dihubungkan dengan jalur data dari IBM PC melalui slot, pin-pin B0 s/d B7 dihubungkan dengan jalur input dari rangkaian latch. Pin Enable (G) dari 74LS245 dihubungkan dengan Y7 dari dekoder, sehingga penyangga ini hanya aktif pada saat

alamat I/O port yang dipilih aktif. Pin DIR dihubungkan dengan jalur IOR dari slot, sehingga saat terjadi proses pembacaan data maka IOR akan aktif (low), dan data akan berpindah dari sisi B ke sisi A.



Gambar 3.11

Rangkaian buffer

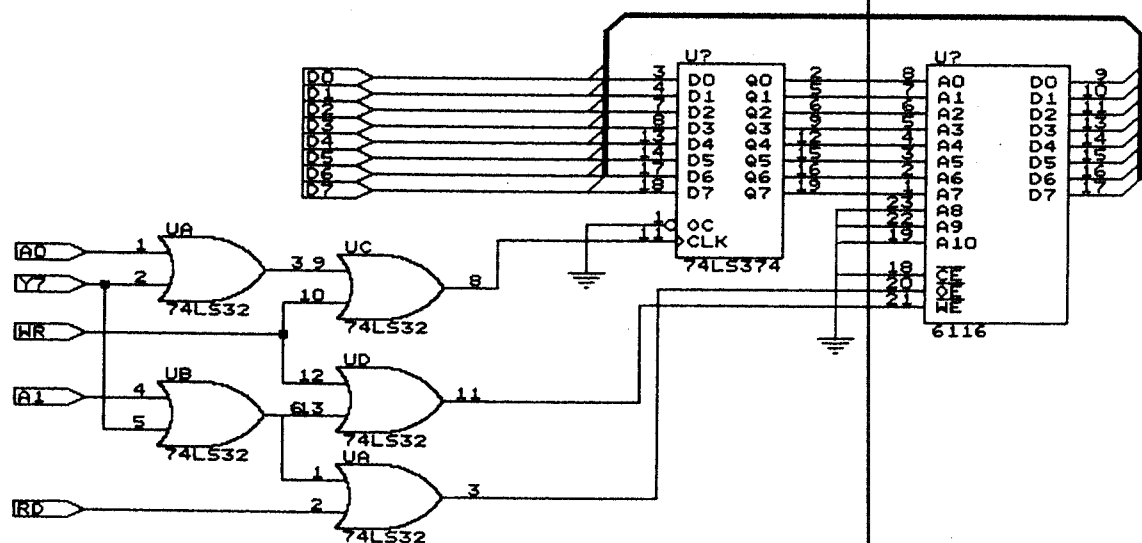
### III.6.5 LATCH

Komponen latch dibutuhkan untuk menahan data yang dikeluarkan oleh I/O port dari IBM PC. Latch yang digunakan tipe 74LS377 yang mempunyai tabel kebenaran seperti pada tabel 3.3. Dari tabel tersebut terlihat bahwa proses latching terjadi pada saat transisi pulsa dari low ke high pada pin clock. Pada saat transisi tersebut maka data output akan sama dengan data input, dan setelah pulsa dari pin clock dalam keadaan high (low) maka data output akan tetap pada keadaan semula meskipun data input berubah. Gambar 3.8 menunjukkan rangkaian latch yang dibuat.

TABEL 3.3

TABEL KEBENARAN LATCH 74LS377

Output Control	Clock	D	Output
L	J	H	H
L	J	L	L
L	L	X	Q
H	X	X	Z <sup>0</sup>



Gambar 3.8

Rangkaian latch

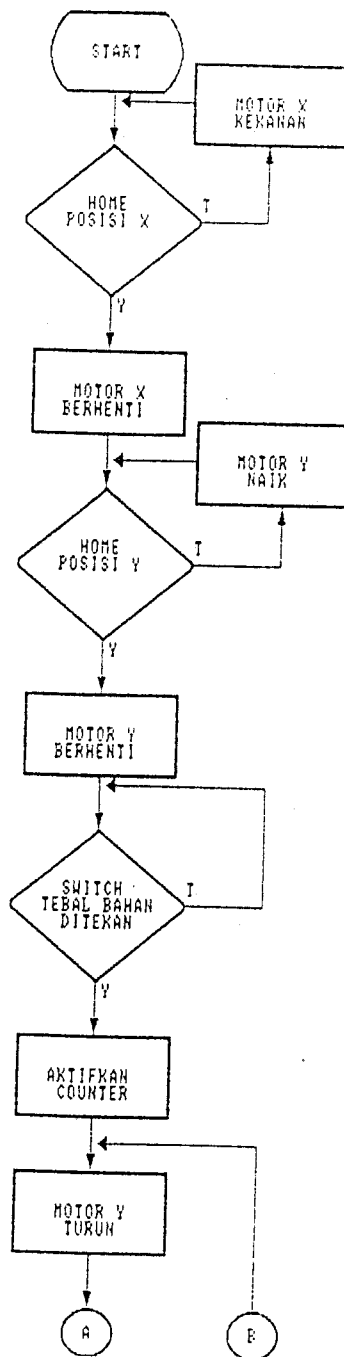
Rangkaian secara keseluruhan dari sistem controller dan interface pengisi data model di RAM diperlihatkan pada lampiran.

### III.7 PERANGKAT LUNAK

Perangkat lunak yang digunakan disini dibagi menjadi dua. Yang pertama adalah perangkat lunak untuk

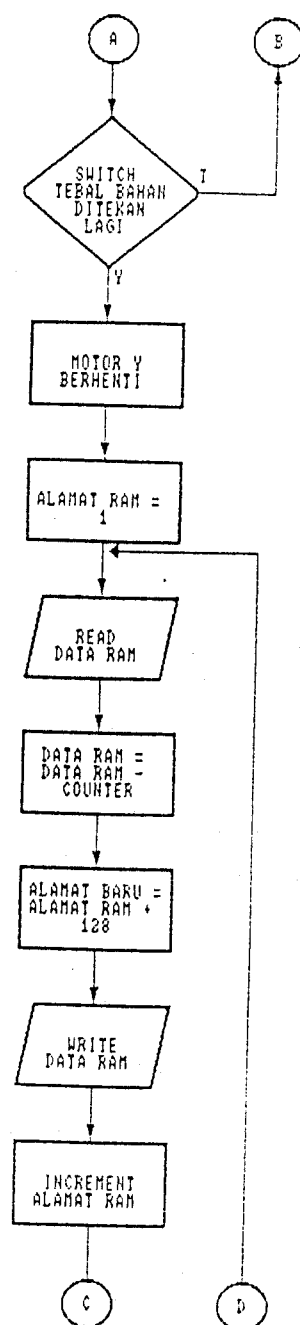
pembuatan model yang dilakukan di IBM PC. Pada perangkat lunak untuk pembuatan model ini digunakan bahasa pascal, suatu bahasa tingkat tinggi yang sudah umum dipakai. Sedang perangkat lunak yang kedua digunakan untuk pengontrolan kerja mesin bubut tersebut. Perangkat lunak yang digunakan adalah bahasa assembly dari mikrokomputer 8035.

Algoritma dari perangkat lunak tersebut diperlihatkan dibawah ini, sedang program lengkapnya dapat dilihat pada lampiran. Pada gambar 3.9a diperlihatkan bagian dari subrutin posisi awal pada pemakanan sumbu X dan pemakanan pada sumbu Y kemudian diteruskan dengan penekanan switch ketebalan bahan. Dan subrutin yang dilakukan setelah penekanan switch tersebut diperlihatkan pada gambar 3.9b. Sedang gambar 3.9c memperlihatkan pemilihan dari pemakanan disini diberikan 3 macam pemakanan yaitu feed A untuk pemakanan tiap 2 mm, feed B untuk pemakanan tiap 3 mm dan feed C untuk pemakanan tiap 4 mm. Sedangkan gambar 3.9d sampai gambar 3.9g memperlihatkan diagram alir dari proses pemakanan oleh pahat terhadap benda kerja sesuai dengan data model yang telah disimpan di RAM.



Gambar 3.9a

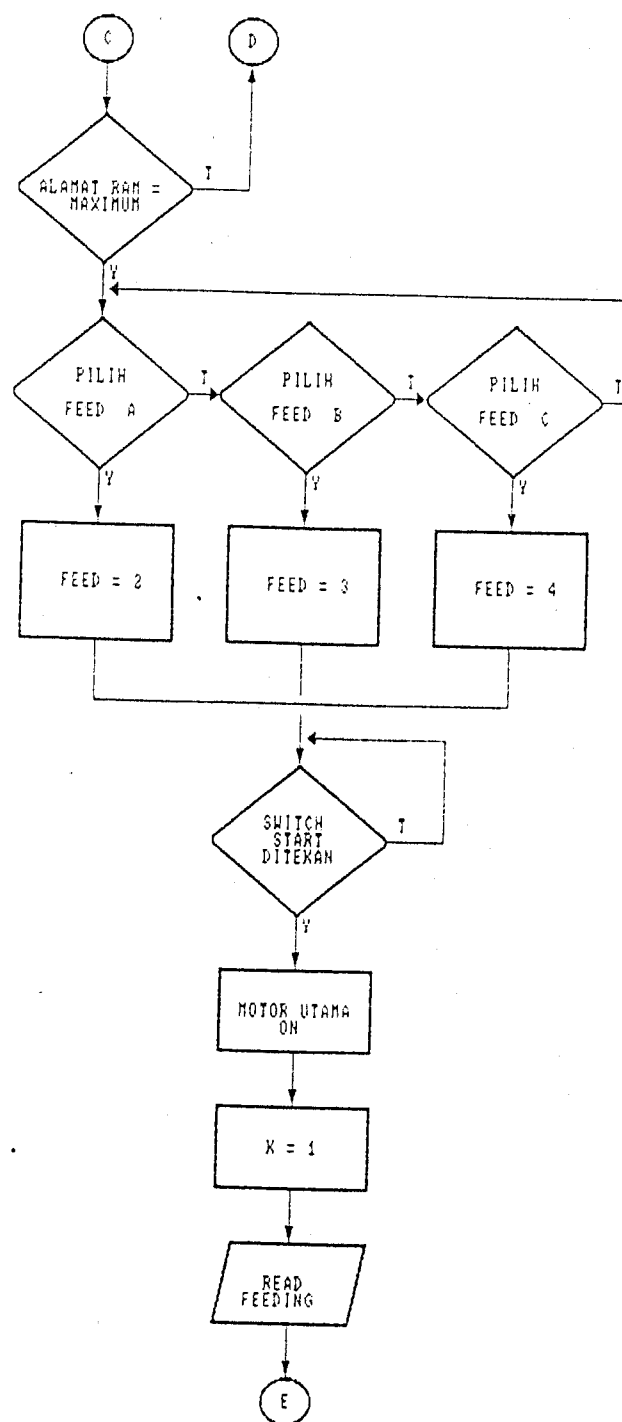
Diagram alir kontrol mesin bubut



Gambar 3.9b

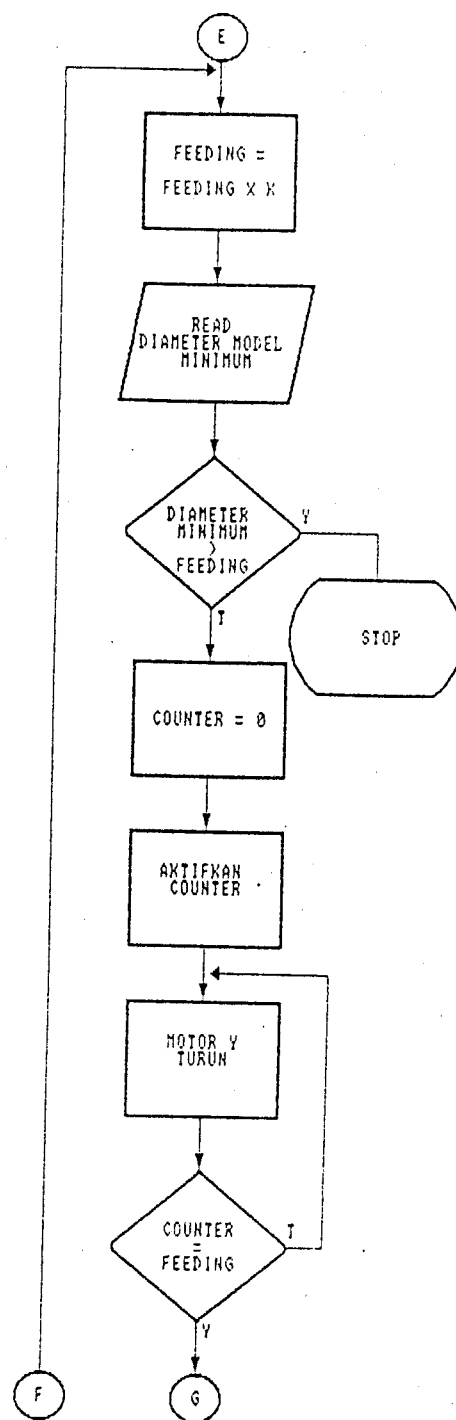
Diagram alir kontrol mesin bubut (lanjutan)





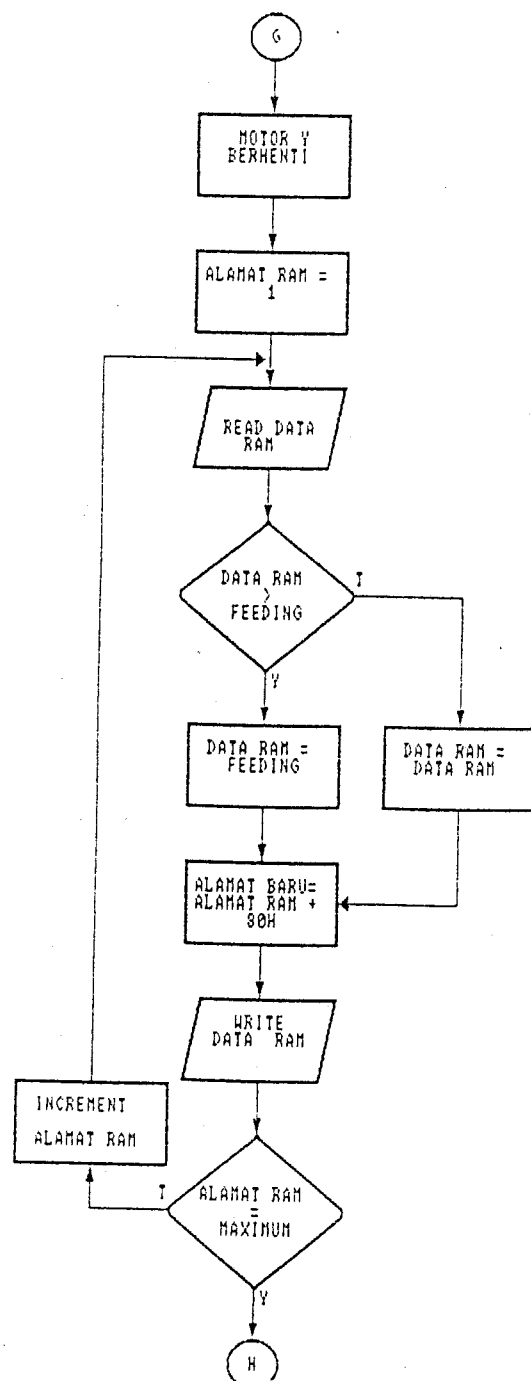
Gambar 3.9c

Diagram alir kontrol mesin bubut (lanjutan)



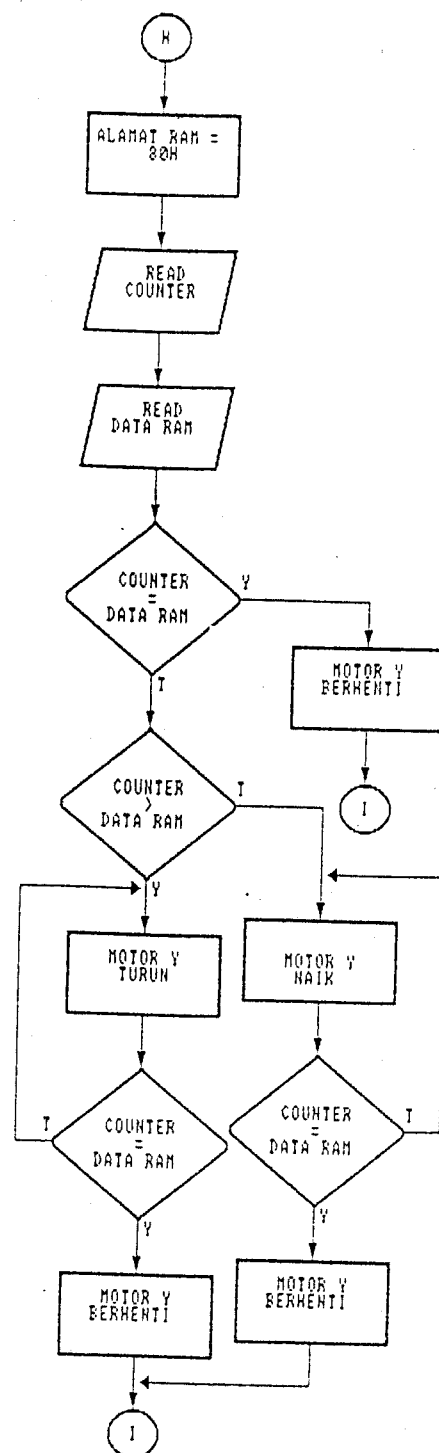
Gambar 3.9d

Diagram alir kontrol mesin bubut (lanjutan)



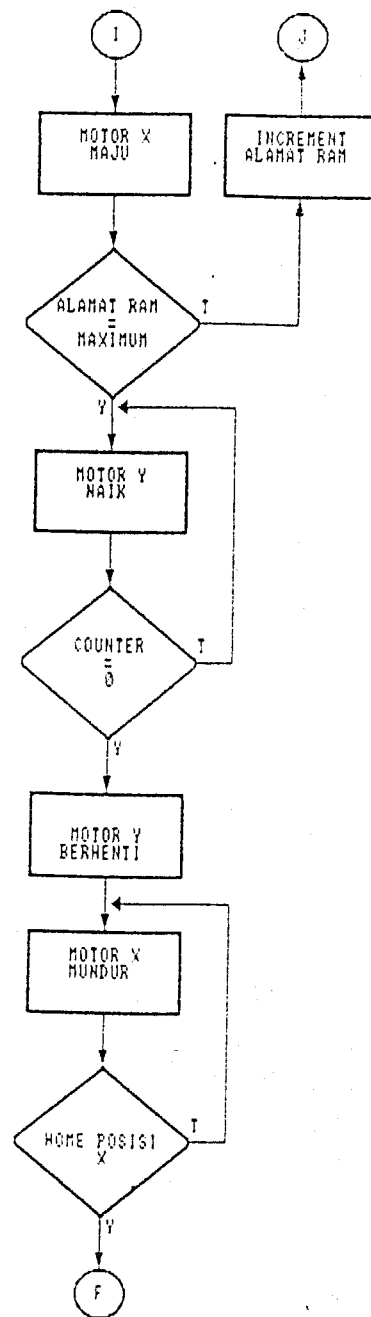
Gambar 3.9e

Diagram alir kontrol mesin bubut (lanjutan)



Gambar 3.9f

Diagram alir kontrol mesin bubut (lanjutan)



Gambar 3.9g

Diagram alir kontrol mesin bubut (lanjutan)

#### 4.1 PENDAHULUAN

Bab ini akan membahas pengukuran dari sinyal PSEN dan sinyal ALE. Kemudian pada bagian selanjutnya akan diberikan foto dari alat yang telah dibuat dan contoh pembuatan model di IBM PC.

#### 4.2 PENGUKURAN SINYAL PSEN

Besar sinyal PSEN dapat dihitung dengan melihat bentuk gelombang dan ukuran pada osiloskop. Dari sinyal pada gambar 4.1 dapat dihitung :

$$\text{time per div.} = 0.5 \mu\text{s}$$

$$\text{volt per div.} = 2$$

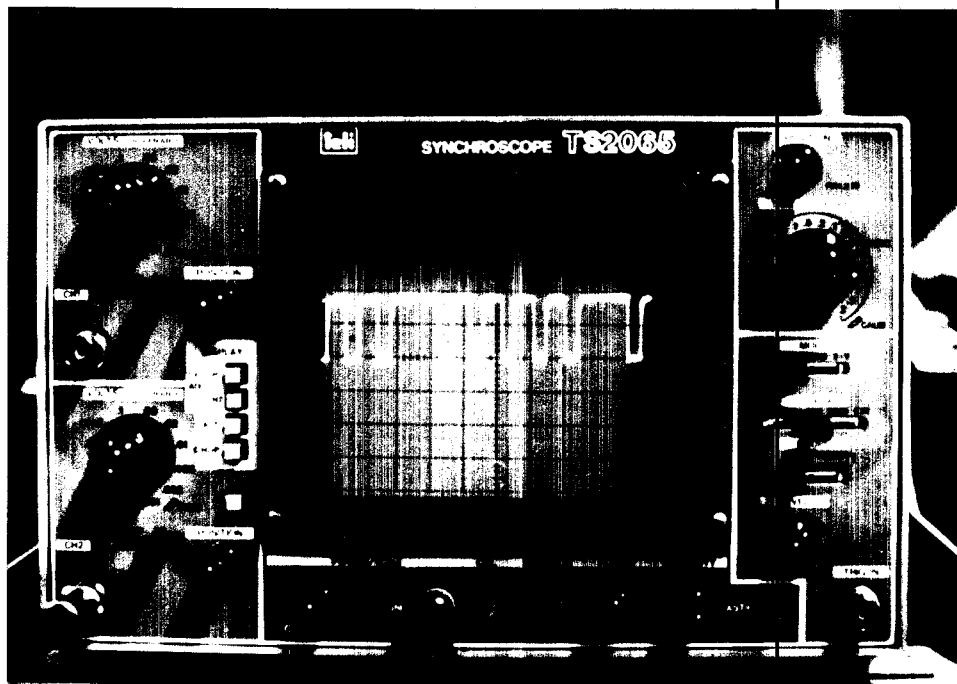
$$T = 3.4 \times 0.5 \mu\text{s}$$

$$= 1.7 \mu\text{s}$$

$$\text{Frekwensi} = 1/T$$

$$= 1 / 1.7 \mu\text{s}$$

$$= 0.58 \text{ MHz}$$



Gambar 4.2

Bentuk gelombang PSEN

#### 4.3 PENGUKURAN SINYAL ALE

Sinyal ALE diukur dengan melihat responnya pada osiloskop. Cara lain dengan membagi lima sinyal CLK. Dari perhitungan sebelumnya didapatkan sinyal CLK sebesar 2.67 MHz. Dari perhitungan didapatkan sinyal ALE sebesar 0.533 MHz (2.67 MHz : 5).

Pengukuran dengan osiloskop didapatkan :

$$\text{time per div.} = 0.5 \mu\text{s}$$

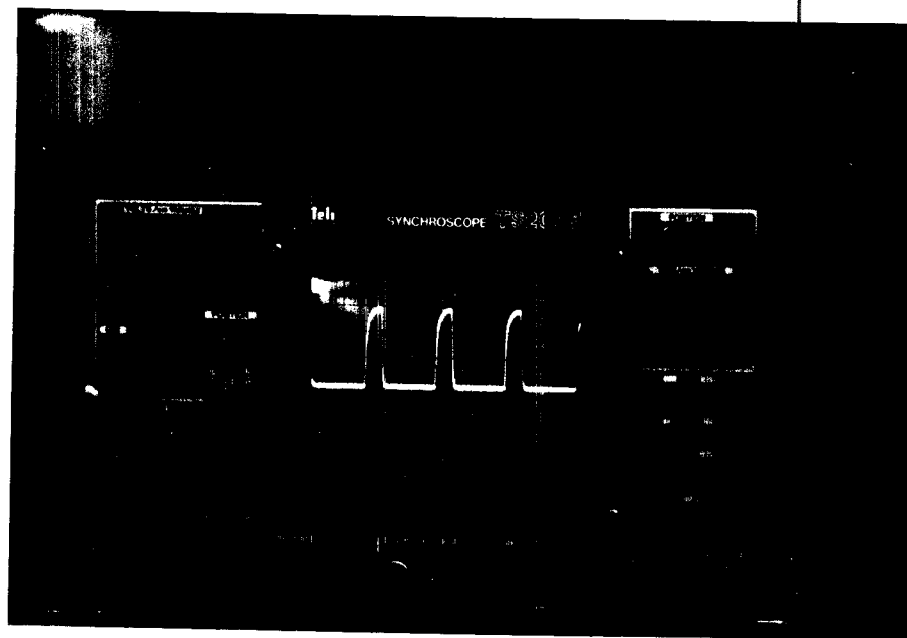
$$\text{Volt per div.} = 2 \text{ Volt}$$

$$T = 3.6 \times 0.5 \mu\text{s}$$

$$= 1.8 \mu\text{s}$$

$$\begin{aligned}
 \text{Frekwensi} &= 1 / T \\
 &= 1 / 1.8 \text{ } \mu\text{s} \\
 &= 0.556 \text{ MHz}
 \end{aligned}$$

Gambar 4.2 menunjukkan bentuk sinyal dari ALE.



Gambar 4.2  
Bentuk sinyal ALE

Foto dari alat yang dibuat diperlihatkan pada gambar 4.3

#### 4.4 CONTOH PEMBUATAN MODEL

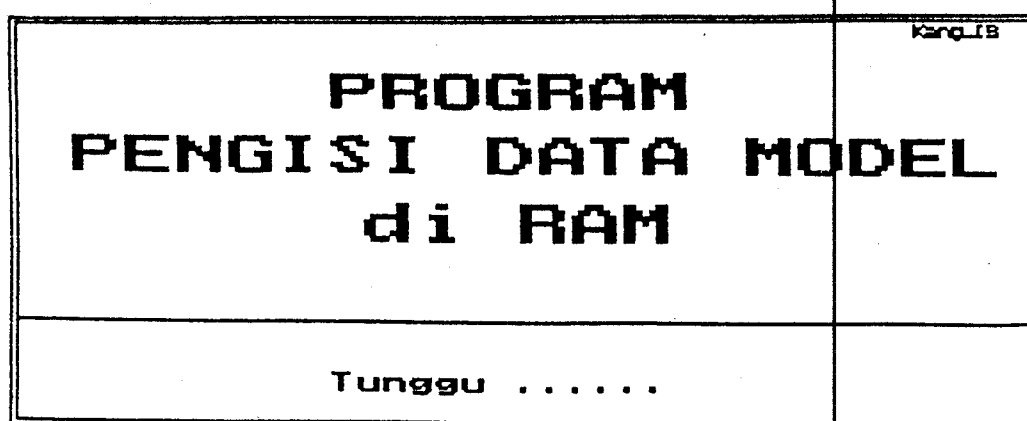
Urutan gambar 4.4 sampai gambar 4.12 menunjukkan pembuatan salah satu contoh model dari bahan kerja yang akan dibubut.








Gambar 4.3

Foto dari alat yang telah dibuat



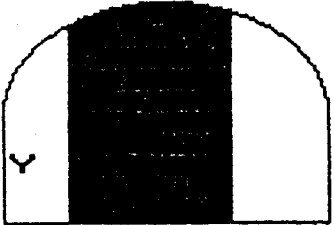
Gambar 4.4

Tampilan utama dari pembuatan model

		
<b>1</b>	<b>2</b>	<b>3</b>
<b>Pilih No = 1</b>		

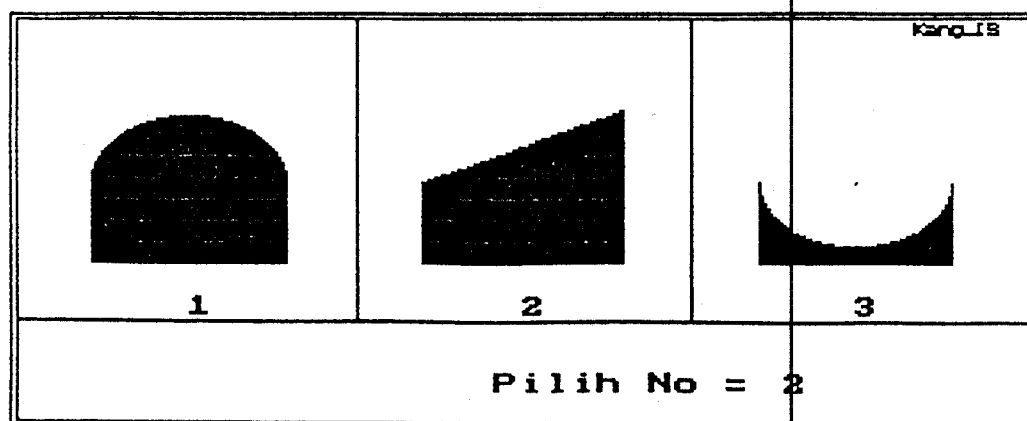
Gambar 4.5

Pemilihan gambar model pertama

	<p align="right"><b>Besar(mm) :</b></p> <p align="right">- . Y = 20</p> <p align="right">- . X1 = 0</p> <p align="right">- . X2 = 0</p> <p align="right">- . R = 20</p>
<b>Lagi (y/t) ? y</b>	

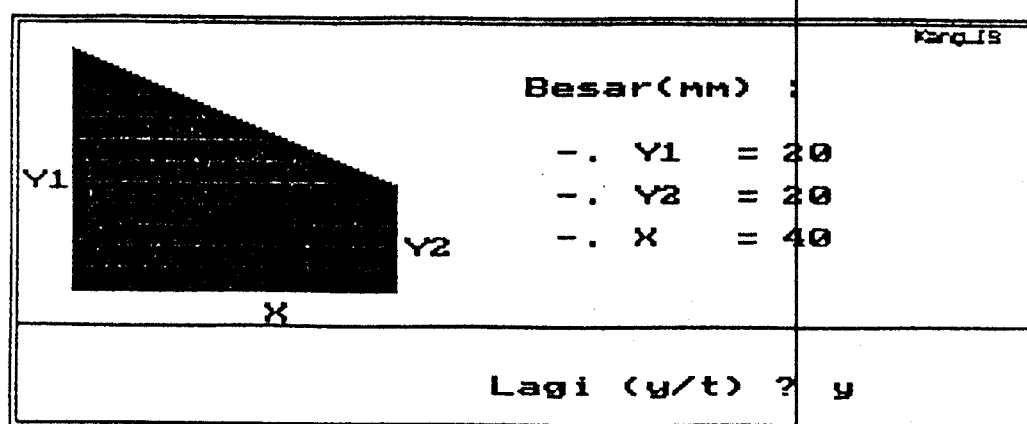
Gambar 4.6

Pengisian parameter model pertama



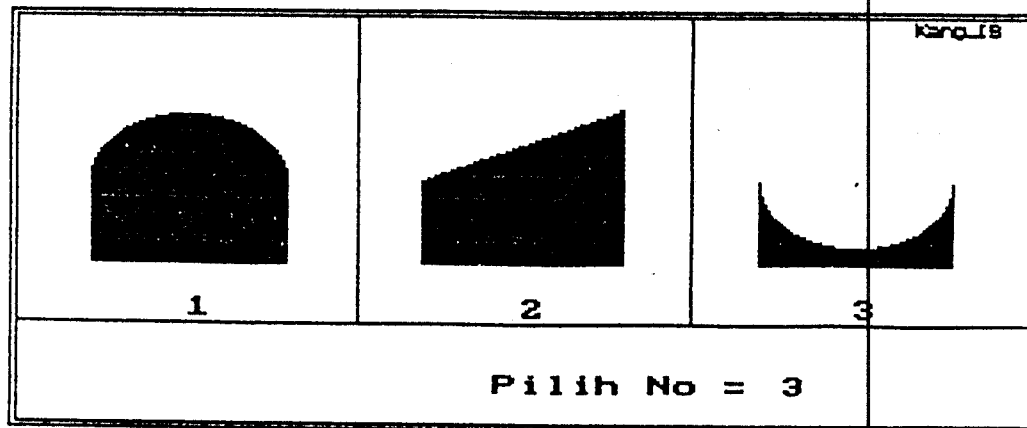
Gambar 4.7

Pemilihan gambar model kedua



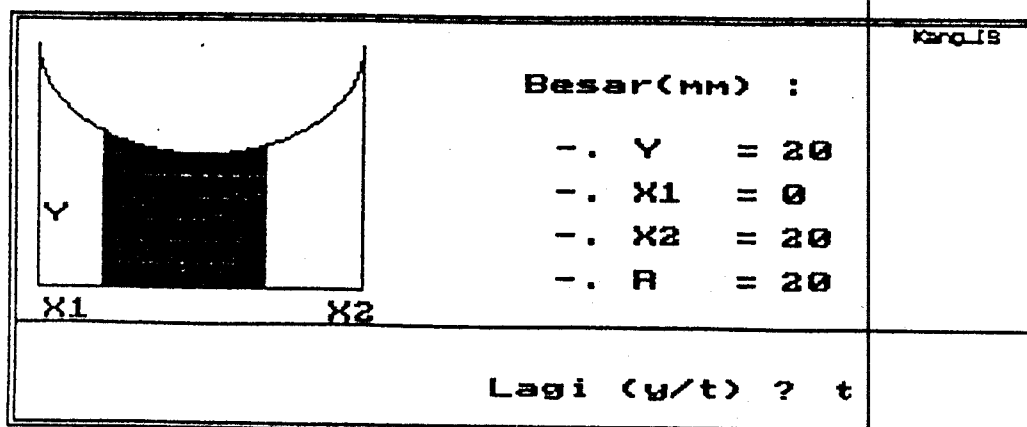
Gambar 4.8

Pengisian parameter model kedua



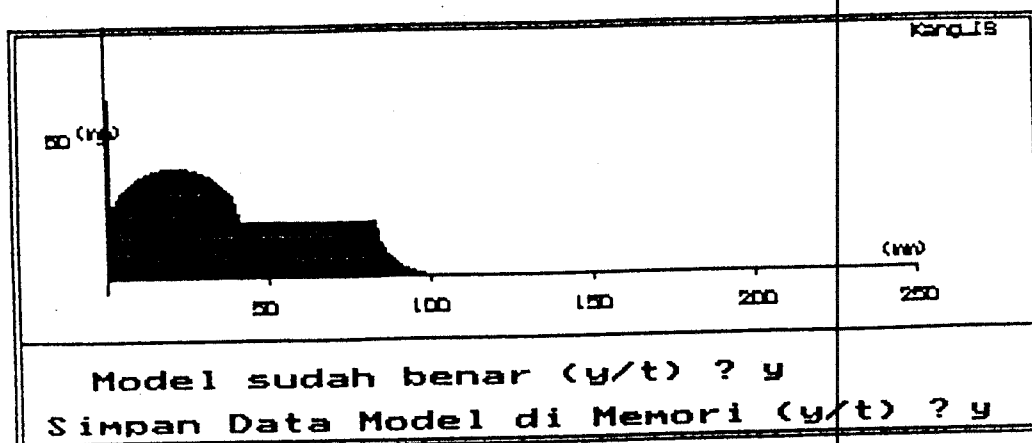
Gambar 4.9

Pemilihan gambar model ketiga



Gambar 4.10

Pengisian parameter model ketiga



Gambar 4.11

Hasil akhir dari model yang telah dibuat

Dari tahap-tahap perencanaan, pembuatan dan pengujian dari sistem ini maka dapat disimpulkan beberapa hal sebagai berikut :

1. Dalam sistem ini penggunaan mikrokomputer 8035 sebagai sarana pengontrol kerja dari mesin bubut sangat menguntungkan. Karena dengan menggunakan komponen ini, tidak dibutuhkan lagi suatu peralatan input-output seperti PPI 8255 karena sudah tersedianya port-port yang dapat diperlakukan sebagai sarana input atau port sebanyak 24 jalur. Disamping itu mikrokomputer ini menyediakan prasarana counter yang dalam perencanaan ini digunakan untuk mengetahui jarak antara benda kerja dengan tool, besar gerak maju untuk pemakanan. Dan prasarana timer yang dalam tugas akhir ini dipergunakan dipergunakan sebagai delay untuk menghilangkan efek debounce pada saat penekanan switch dan timer yang berhubungan dengan interrupt. Mikrokomputer ini tidak efisien lagi jika banyak dibutuhkan data memori sebagai penyimpan data sementara. Karena internal RAM yang disediakan hanya sebesar 64 byte, sedang external RAM bisa diperluas sampai 256 byte.

2. Proses pengisian data model di RAM dilakukan di IBM PC disebabkan banyak operasi matematika yang harus dilakukan dalam pembuatan model. Sebagai misal untuk mendapatkan besar busur dari setengah lingkaran dibutuhkan operasi perkalian, akar dan kwadrat. Kerugian dari pemakaian IBM PC sebagai prasarana pengisi data model adalah ketergantungannya proses pengerjaan terhadap IBM PC. Untuk mengatasi masalah tersebut disediakan pilihan pengerjaan secara manual.

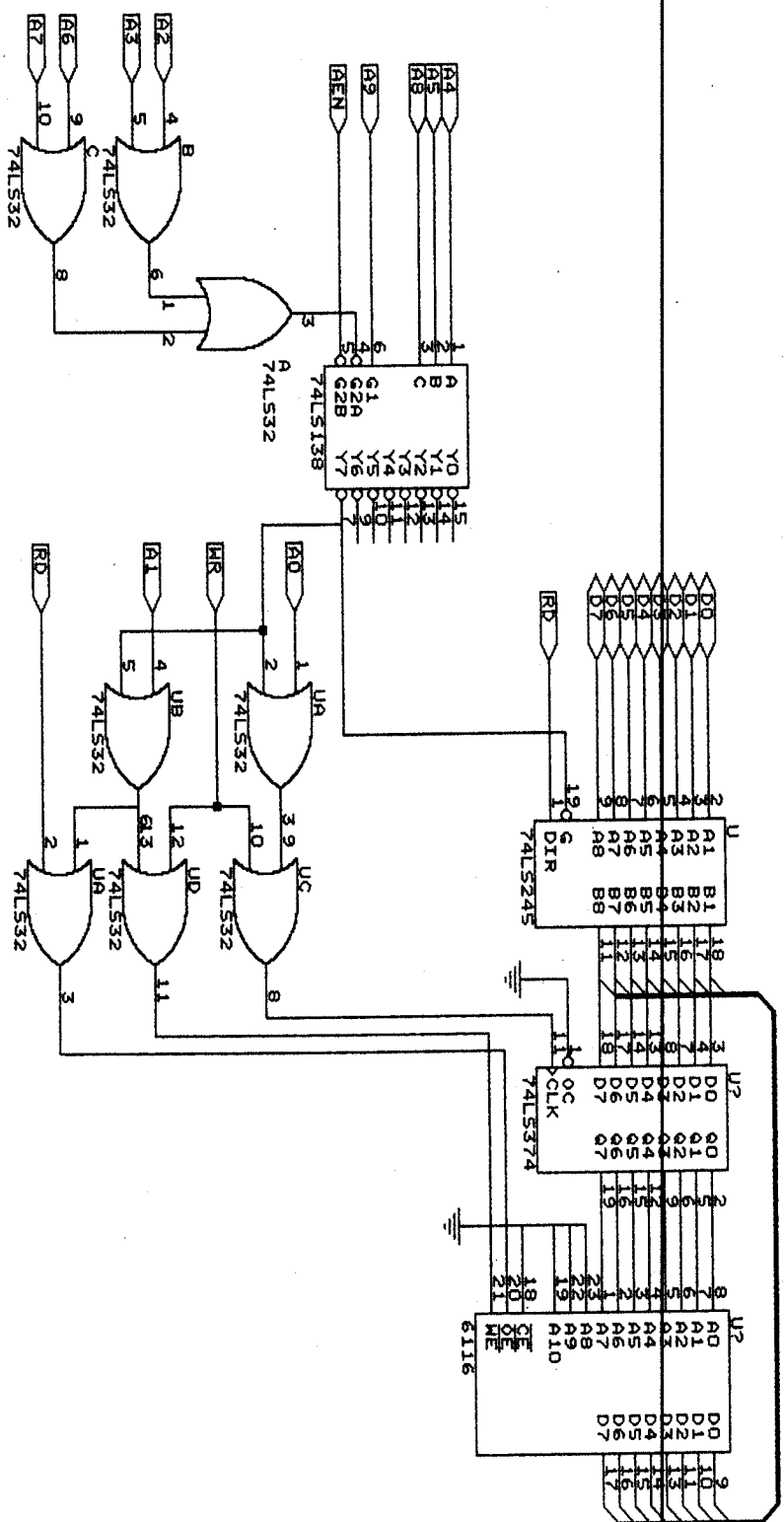
3. Ketelitian dalam pembuatan mekanik mempengaruhi hasil dari alat yang dibuat. Pada perencanaan mesin bubut yang dibuat ini untuk kenaikan 1 pulsa counter, motor akan menggerakkan tool sebesar 1 mm. Sebenarnya alat yang dibuat masih mampu untuk masih mampu untuk menggerakkan tool sebesar 0.2 mm tiap satu pulsa, tapi berhubung mekanik yang dibuat ketelitiannya sangat jelek maka hal tersebut tidak dilakukan.

## DAFTAR PUSTAKA

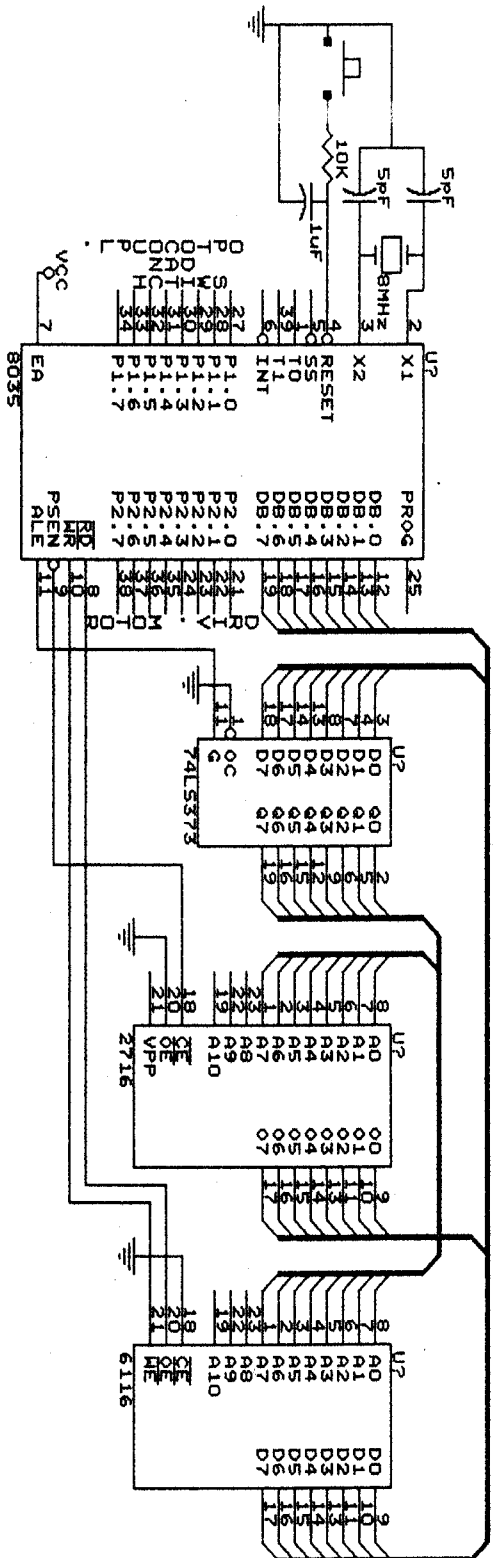
1. ...., Single Chip 8-Bit Microcontrollers, User Manual, Philips Electronics Component Materials, 1986.
2. Borland., Turbo Pascal Professional Reference Guide., Borland International., Scotts Valley, 1989.
3. Coffron, James W., Z80 Application., Sybex Inc., 1983.
4. Eggebrecht, Lewis C., Interfacing to the IBM Personal Computer. Indianapolis : Howard W. Sams & Co., 1st Ed., 1987.
5. Gerling, Heinrich., All About Machine Tools, Wiley Eastern Limited. New Delhi, 1965.
6. Hall, Douglas G, Microprocessor And Interfacing Programming And Hardware, Mc Graw Hill Book Company, New York, 1988.
7. Hartono Partoharsodjo., Tuntunan Praktis Pemrograman Bahasa Pascal., PT Elex Media Komputindo., Jakarta, 1989
8. Horowitz, Paul., Ignatius Hartono., Seni Dan Disain Elektronika., Elex Media Komputindo, Jakarta, 1985.



9. Maloney, Timothy J, Industrial Solid-State Electronics, Prentice Hall, New Jersey, 1979.
10. Texas Instruments, TTL Data Book For Design Engineers, Texas Instruments Inc. Dallas, Texas, 1980.



Title	Interface pengisi data model di RAM
Size Document Number	REV
A	
Date: January 1, 1980	Sheet 1 of 1



Title		Rangkaian kontroler utama	
Size Document Number		2	
Date: January 1, 1980		Sheet of	
REV			

intel

PRELIMINARY

## 8048L SPECIAL LOW POWER CONSUMPTION SINGLE COMPONENT 8-BIT MICROCOMPUTER

- Typical Power Consumption 100mW
- Typical Standby Power 10mW  
V<sub>DD</sub> minimum of 2.2V
- 8-Bit CPU, ROM, RAM, I/O In Single Package
- 4.17  $\mu$ sec Instruction Cycle.  
All Instructions 1 or 2 Cycles.
- Over 90 Instructions: 70% Single Byte
- 1K x 8 ROM
- 64 x 8 RAM
- 27 I/O Lines
- Interval Timer/Event Counter
- Easily Expandable Memory and I/O
- Compatible with 8080/8085 Series Peripherals
- Two Single Level Interrupts

The Intel® 8048L is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's advanced N-channel silicon gate HMOS process, using special techniques to reduce operating and standby power consumption. The 8048L contains a 1K X 8 program memory, a 64 X 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on-board oscillator and clock circuits. For systems that require extra capability the 8048L can be expanded using standard memories and MCS-80®/MCS-85® peripherals. The 8048L can be used with external ROM and RAM.

To reduce development problems to a minimum and provide maximum flexibility, a logically and functionally pin compatible version of the 8048L with UV-erasable user-programmable EPROM program memory is available. The 8748 will emulate the 8048L with greater power and other minor differences.

This microcontroller is designed to be an efficient controller as well as an arithmetic processor. The 8048L has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over two bytes in length.

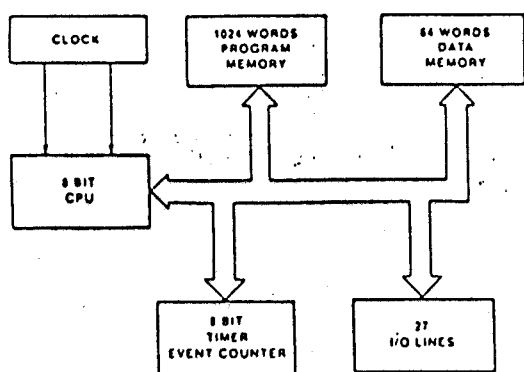


Figure 1.  
8048L Block Diagram

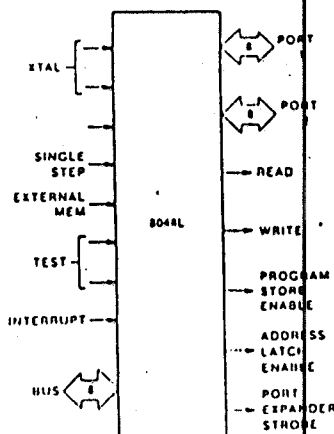


Figure 2.  
8048L Logic Symbol

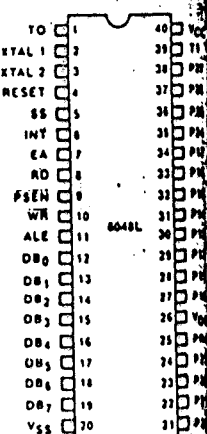


Figure 3.  
8048L Pin Configuration

intel

8048L

PRELIMINARY

## PIN DESCRIPTION

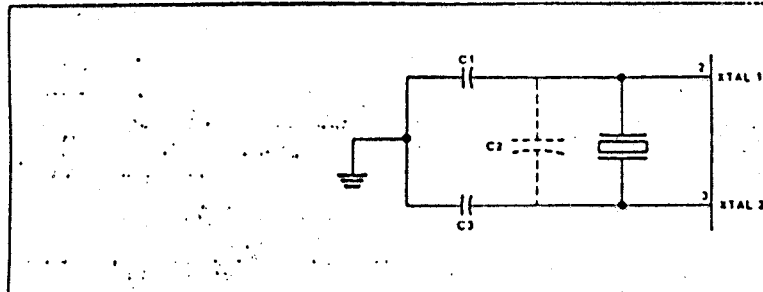
Designation	Pin =	Function	Designation	Pin =	Function
V <sub>SS</sub>	20	Circuit GND potential			testable with conditional jump instruction. (Active low)
V <sub>DD</sub>	26	Low power standby pin			
V <sub>CC</sub>	40	Main power supply; +5V during operation.	$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device.
PROG	25	Output strobe for 8243 I/O expander.			Used as a read strobe to external data memory. (Active low)
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.			
P20-27 Port 2	21-24	8-bit quasi-bidirectional port.	$\overline{RESET}$	4	Input which is used to initialize the processor. (Active low) (Non TTL V <sub>IH</sub> )
	35-38	P20-P23 contain the four high order program counter bits during an external program memory latch and serve as a 4-bit I/O expander bus for 8243.	$\overline{WR}$	10	Output strobe during a bus write. (Active low)
DB0-DB7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$ , $\overline{WR}$ strobes. The port can also be statically latched.	ALE	11	Used as write strobe to external data memory.
		Contains the 8 low order program counter bits during an external program memory latch, and receives the addressed instruction under the control of $\overline{PSEN}$ . Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$ , and $\overline{WR}$ .			Address latch enable. This signal occurs once during each cycle and is useful as a clock output.
			$\overline{PSEN}$	9	The negative edge of ALE strobes address into external data and program memory.
			$\overline{SS}$	5	Program store enable. This output occurs only during a fetch to external program memory. (Active low)
10	1	Input pin testable using the conditional transfer instructions JTO and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.	EA	7	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
11	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.			External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
INT	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also	XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V <sub>IH</sub> )
			XTAL2	3	Other side of crystal input.

intel

8048L

PRELIMINARY

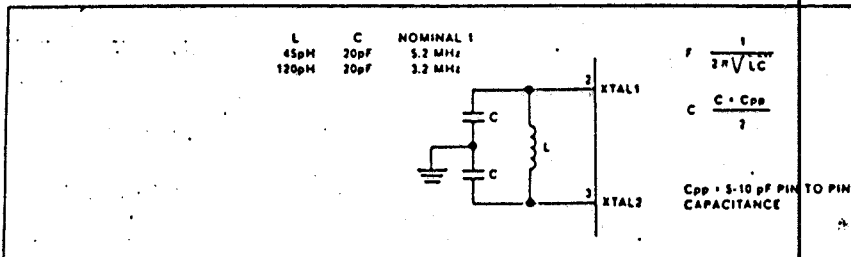
# CRYSTAL OSCILLATOR MODE



$C1 = 8pF \pm 1/2pF + STRAY < 8pF$   
 $C2 = CRYSTAL + STRAY < 8pF$   
 $C3 = 20pF \pm 1pF + STRAY < 5pF$

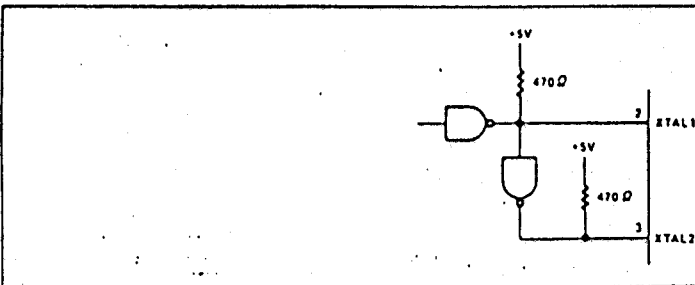
CRYSTAL SERIES RESISTANCE SHOULD BE LESS THAN  $75\Omega$  AT 6 MHz LESS THAN  $180\Omega$  AT 3.6MHz

# LC OSCILLATOR MODE



EACH C SHOULD BE APPROXIMATELY 20pF, INCLUDING STRAY CAPACITANCE

# DRIVING FROM EXTERNAL SOURCE



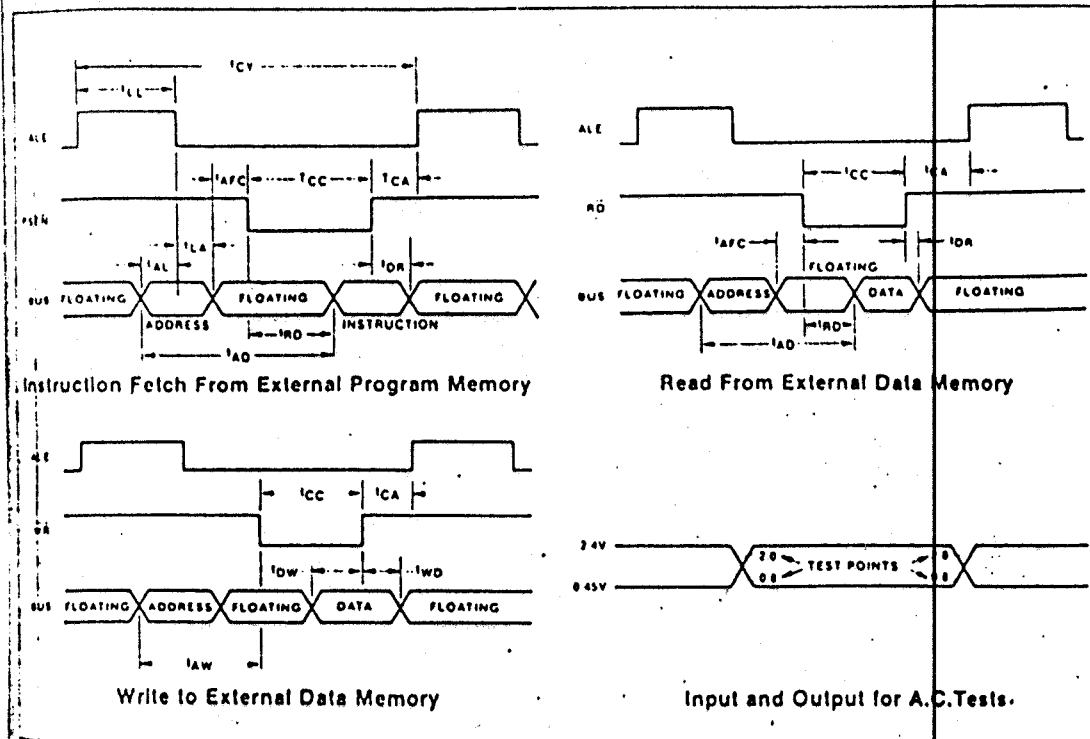
XTAL 1 MUST BE HIGH 35-65% OF THE PERIOD AND XTAL 2 MUST BE HIGH 35-65% OF THE PERIOD. RISE AND FALL TIMES MUST NOT EXCEED 20ns.

intel

8048L

PRELIMINARY

# WAVEFORMS



A.C. CHARACTERISTICS  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = V_{DD} = 5V \pm 10\%$ ,  $V_{SS} = 0V$

Symbol	Parameter			Unit	Conditions (Note 1)
		Min.	Max.		
$t_{LL}$	ALE Pulse Width	600		ns	
$t_{AL}$	Address Setup to ALE	150		ns	
$t_{LA}$	Address Hold from ALE	80		ns	
$t_{CC}$	Control Pulse Width (PSEN, RD, WR)	1500		ns	
$t_{DW}$	Data Setup before WR	640		ns	
$t_{WD}$	Data Hold After WR	120		ns	$C_L = 20\text{pF}$
$t_{CY}$	Cycle Time	4.17	15.0	$\mu\text{s}$	
$t_{OR}$	Data Hold	0	200	ns	
$t_{RD}$	PSEN, RD to Data In		750	ns	
$t_{AW}$	Address Setup to WR	260		ns	
$t_{AD}$	Address Setup to Data In		1450	ns	
$t_{AFC}$	Address Float to RD, PSEN	0		ns	
$t_{CA}$	Control Pulse to ALE	20		ns	

Note 1: Control outputs:  $C_L = 80\text{ pF}$   
 BUS Outputs:  $C_L = 150\text{ pF}$

intel

8048L

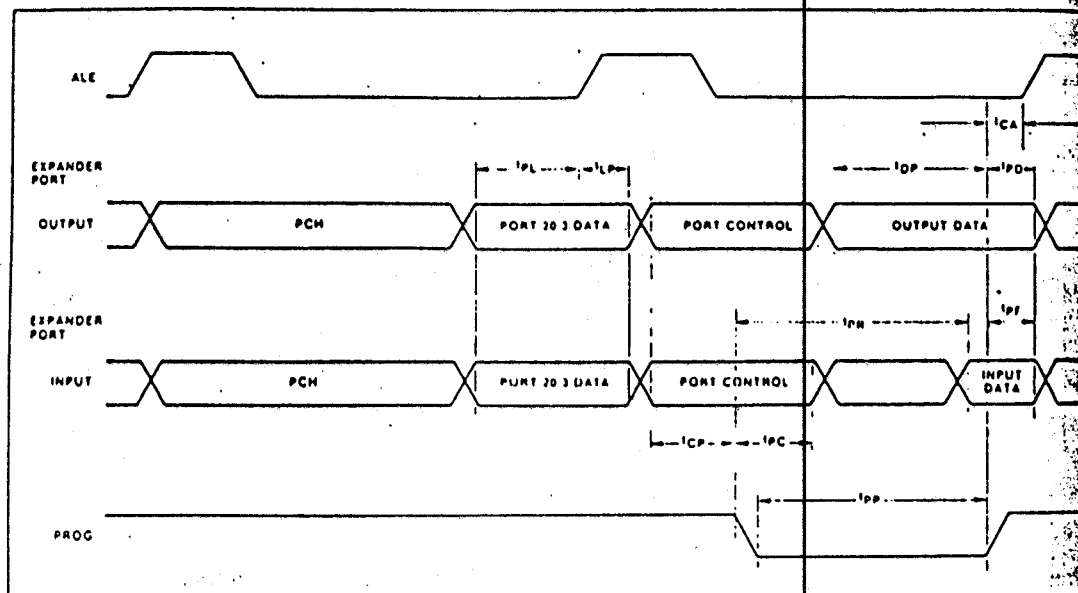
PRELIMINARY

## A.C. CHARACTERISTICS (PORT 2 TIMING)

 $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5\text{V} \pm 10\%$ ,  $V_{SS} = 0\text{V}$  $TCY = 4.17 \mu\text{S}$ 

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
$t_{CP}$	Port Control Setup Before Falling Edge of PROG	185		ns	
$t_{PC}$	Port Control Hold After Falling Edge of PROG	160		ns	
$t_{PR}$	PROG to Time P2 Input Must Be Valid		1.35	$\mu\text{s}$	
$t_{PF}$	Input Data Hold Time	0	250	ns	
$t_{DP}$	Output Data Setup Time	420		ns	
$t_{PD}$	Output Data Hold Time	110		ns	
$t_{PP}$	PROG Pulse Width	2.0		$\mu\text{s}$	
$t_{PL}$	Port 2 I/O Data Setup	585		ns	
$t_{LP}$	Port 2 I/O Data Hold	250		ns	

## PORT 2 TIMING

BUS TIMING AS A FUNCTION OF  $TCY$ 

SYMBOL	FUNCTION OF $TCY$
$T_{LL}$	7/30 $TCY$ MIN
$T_{AL}$	2/15 $TCY$ MIN
$T_{LA}$	1/15 $TCY$ MIN
$T_{CC(1)}$	1/2 $TCY$ MIN
$T_{CC(2)}$	2/5 $TCY$ MIN
$T_{DW}$	13/30 $TCY$ MIN
$T_{WD}$	1/15 $TCY$ MIN
$T_{DR}$	0 MIN

$T_{CC(1)} : \overline{RD}/\overline{WR}$   
 $T_{CC(2)} : \overline{PSEN}$

SYMBOL	FUNCTION OF $TCY$
$T_{RD(1)}$	2/5 $TCY$ MAX
$T_{RD(2)}$	3/10 $TCY$ MAX
$T_{AW}$	1/3 $TCY$ MIN
$T_{AD(1)}$	11/15 $TCY$ MAX
$T_{AD(2)}$	8/15 $TCY$ MAX
$T_{AFC(1)}$	2/15 $TCY$ MIN
$T_{AFC(2)}$	1/30 $TCY$ MIN
$T_{CA(1)}$	1/15 $TCY$ MIN
$T_{CA(2)}$	2/15 $TCY$ MIN

$T_{RD(1)} : \overline{RD}$   
 $T_{RD(2)} : \overline{PSEN}$   
 $T_{AD(1)} : \overline{RD}$   
 $T_{AD(2)} : \overline{PSEN}$   
 $T_{AFC(1)} : \overline{RD}$   
 $T_{AFC(2)} : \overline{PSEN}$   
 $T_{CA(1)} : \overline{RD}, \overline{WR}$   
 $T_{CA(2)} : \overline{PSEN}$

\* APPROXIMATE VALUES NOT INCLUDING GATE DELAYS.



intel

8048L

PRELIMINARY

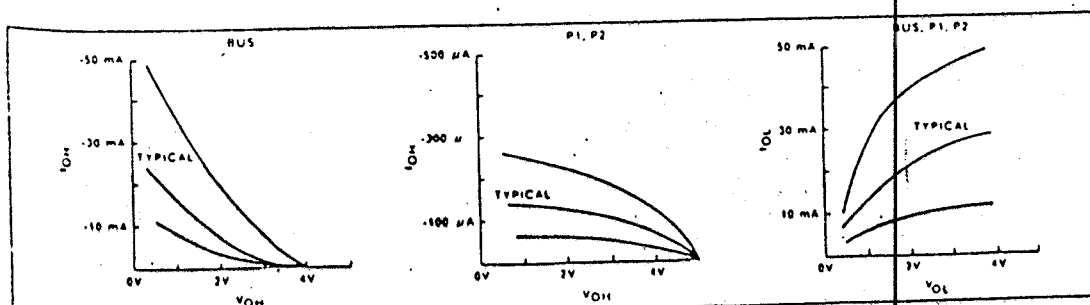
## ABSOLUTE MAXIMUM RATINGS\*

Ambient Temperature Under Bias ..... 0°C to 70°C  
 Storage Temperature ..... -65°C to +125°C  
 Voltage On Any Pin With Respect  
 to Ground ..... -0.5V to +7V  
 Power Dissipation ..... 1.5 Watt

\* COMMENT Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. AND OPERATING CHARACTERISTICS  $T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = V_{DD} = 5V \pm 10\%$ ,  $V_{SS} = 0V$ 

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
$V_{IL}$	Input Low Voltage (All Except $\overline{\text{RESET}}$ , X1, X2)	-0.5		.8	V	
$V_{IL1}$	Input Low Voltage ( $\overline{\text{RESET}}$ , X1, X2)	-0.5		.6	V	
$V_{IH}$	Input High Voltage (All Except XTAL1, XTAL2, $\overline{\text{RESET}}$ )	2.0		$V_{CC}$	V	
$V_{IH1}$	Input High Voltage (X1, X2, $\overline{\text{RESET}}$ )	3.8		$V_{CC}$	V	
$V_{OL}$	Output Low Voltage (BUS)			.45	V	$V_{OL} = 2.0 \text{ mA}$
$V_{OL1}$	Output Low Voltage ( $\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{PSEN}}$ , ALE)			.45	V	$I_{OL} = 1.8 \text{ mA}$
$V_{OL2}$	Output Low Voltage (PROG)			.45	V	$I_{OL} = 1.0 \text{ mA}$
$V_{OL3}$	Output Low Voltage (All Other Outputs)			.45	V	$I_{OL} = 1.6 \text{ mA}$
$V_{OH}$	Output High Voltage (BUS)	2.4			V	$I_{OH} = 400 \mu\text{A}$
$V_{OH1}$	Output High Voltage ( $\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{PSEN}}$ , ALE)	2.4			V	$I_{OH} = 100 \mu\text{A}$
$V_{OH2}$	Output High Voltage (All Other Outputs)	2.4			V	$I_{OH} = 40 \mu\text{A}$
$I_{L1}$	Input Leakage Current (T1, INT)			$\pm 10$	$\mu\text{A}$	$V_{SS} \leq V_{IN} \leq V_{CC}$
$I_{L11}$	Input Leakage Current (P10-P17, P20-P27, EA, $\overline{\text{SS}}$ )			-500	$\mu\text{A}$	$V_{SS} + .45 \leq V_{IN} \leq V_{CC}$
$I_{LO}$	Output Leakage Current (BUS, TO) (High Impedance State)			$\pm 10$	$\mu\text{A}$	$V_{SS} + .45 \leq V_{IN} \leq V_{CC}$
$I_{DD}$	$V_{DD}$ Supply Current		2	4	mA	
$I_{DD} + I_{CC}$	Total Supply Current		20	40	mA	
$V_{DD}$	Ram Standby Pin Voltage	2.2		5.5	V	Standby Mode, $\text{Reset} \leq 0.6V$



## INSTRUCTION SET

Accumulator			
Mnemonic	Description	Bytes	Cycles
ADD A, R	Add register to A	1	1
ADD A, @R	Add data memory to A	1	1
ADD A, # data	Add immediate to A	2	2
ADDC A, R	Add register with carry	1	1
ADDC A, @R	Add data memory with carry	1	1
ADDC A, # data	Add immediate with carry	2	2
ANL A, R	And register to A	1	1
ANL A, @R	And data memory to A	1	1
ANL A, # data	And immediate to A	2	2
ORL A, R	Or register to A	1	1
ORL A, @R	Or data memory to A	1	1
ORL A, # data	Or immediate to A	2	2
XRL A, R	Exclusive or register to A	1	1
XRL A, @R	Exclusive or data memory to A	1	1
XRL A, # data	Exclusive or immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1

Input/Output			
Mnemonic	Description	Bytes	Cycles
IN A, P	Input port to A	1	2
OUTL P, A	Output A to port	1	2
ANL P, # data	And immediate to port	2	2
ORL P, # data	Or immediate to port	2	2
INS A, BUS	Input BUS to A	1	2
OUTL BUS, A	Output A to BUS	1	2
ANL BUS, # data	And immediate to BUS	2	2
ORL BUS, # data	Or immediate to BUS	2	2
MOVD A, P	Input expander port to A	1	2
MOVD P, A	Output A to expander port	1	2
ANLD P, A	And A to expander port	1	2
ORLD P, A	Or A to expander port	1	2

Registers			
Mnemonic	Description	Bytes	Cycles
INC R	Increment register	1	1
INC @R	Increment data memory	1	1
DEC R	Decrement register	1	1

Branch			
Mnemonic	Description	Bytes	Cycles
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ R, addr	Decrement register and skip	2	2
JC addr	Jump on carry = 1	2	2
JNC addr	Jump on carry = 0	2	2
JZ addr	Jump on A zero	2	2
JNZ addr	Jump on A not zero	2	2
JTO addr	Jump on TO = 1	2	2
JNTO addr	Jump on TO = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 = 1	2	2
JF1 addr	Jump on F1 = 1	2	2
JTF addr	Jump on timer flag	2	2
JN1 addr	Jump on INT = 0	2	2
JBb addr	Jump on accumulator bit	2	2

Subroutine			
Mnemonic	Description	Bytes	Cycles
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2

Flags			
Mnemonic	Description	Bytes	Cycles
CLR C	Clear carry	1	1
CPL C	Complement carry	1	1
CLR F0	Clear flag 0	1	1
CPL F0	Complement flag 0	1	1
CLR F1	Clear flag 1	1	1
CPL F1	Complement flag 1	1	1

Data Moves			
Mnemonic	Description	Bytes	Cycles
MOV A, R	Move register to A	1	1
MOV A, @R	Move data memory to A	1	1
MOV A, # data	Move immediate to A	2	2
MOV R, A	Move A to register	1	1
MOV @R, A	Move A to data memory	1	1
MOV R, # data	Move immediate to register	2	2
MOV @R, # data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, R	Exchange A and register	1	1
XCH A, @R	Exchange A and data memory	1	1
XCHD A, @R	Exchange nibble of A and register	1	1
MOVX A, @R	Move external data memory to A	1	2
MOVX @R, A	Move A to external data memory	1	2
MOVP A, @A	Move to A from current page	1	2
MOVPJ A, @A	Move to A from page 3	1	2

Timer/Counter			
Mnemonic	Description	Bytes	Cycles
MOV A, T	Load timer/counter	1	1
MOV T, A	Load timer/counter	1	1
STRT T	Start timer	1	1
STRT CNT	Start counter	1	1
STOP TCNT	Stop timer/counter	1	1
EN TCNTI	Enable timer/counter interrupt	1	1
DIS TCNTI	Disable timer/counter interrupt	1	1

Control			
Mnemonic	Description	Bytes	Cycles
EN I	Enable external interrupt	1	1
DIS I	Disable external interrupt	1	1
SEL RD0	Select register bank 0	1	1
SEL RD1	Select register bank 1	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
ENT 0 CLK	Enable clock output on T0	1	1

NOP			
Mnemonic	Description	Bytes	Cycles
NOP	No operation	1	1

```

{*****
*          PROGRAM PENYIMPANAN DATA          *
*          MODEL DI RAM                       *
*
*          Oleh :                             *
*          ISRAMSYAH                          *
*          2852200383                         *
*****}

```

```

uses dos,graph,crt;

```

```

const

```

```

    Alamat          = $0332;
    DtaRam           = $0331;

```

```

var

```

```

    Ex1,
    Ex2,
    Er,
    Ye1,
    Ye2,
    Ex,
    Ye,
    AtaString       : string;
    X1,
    X2,
    Y,
    R,
    akhir,
    Yreal1,
    pos,
    Yreal           : real;
    DataBuffer      : array[1..2000] of real;
    DataBaru        : array[1..2000] of real;
    sumbuY,
    sumbuX           : Real;
    lstring          : String[3];
    Driv,
    Mode,
    GraphMode,
    i                : integer;
    Dia,
    DiaBaru,
    Data             : real;
    Counter,
    K,
    AmbilData,
    Lipat,
    Makan,
    Feed,
    maksi,ki         : integer;
    Y1,
    Y2,
    X,
    kurang,

```

```

sudut,
bulat,
Mini,
teta          : real;
maksim,
ata,
DataRAM       : Byte;
lagi,
Simpan,
Model,
pilih,
pilih,
pilih,
tekan        : char;

```

```

{*****}
{          PROCEDURE PEMBUATAN GAMBAR MODE_          }
{*****}
procedure MenuUtama;
begin
  clearDevice;
  SetGraphMode(Mode);
  SetTextStyle(SmallFont, Horizdir, 3);
  OutTextXY(280, 20, 'Kang_1S');
  delay(1000);
  Rectangle(3, 20, 317, 170);
  Rectangle(1, 18, 319, 172);
  SetTextStyle(DefaultFont, HorizDir, 0);
  setlinestyle(solidln, 0, normwidth);
  line(3, 133, 317, 133);
  line(108, 20, 108, 133);
  line(211, 20, 211, 133);

  arc(56, 81, 0, 180, 30);
  line(26, 81, 26, 111);
  line(26, 111, 86, 111);
  line(86, 81, 86, 111);
  putpixel(56, 81, 1);
  Floodfill(56, 82, 3);
  moveto(56, 124); outtext('1');

  moveto(128, 81);
  lineto(190, 55);
  lineto(190, 111);
  lineto(128, 111);
  lineto(128, 81);
  floodfill(140, 80, 3);
  moveto(159, 124); outtext('2');

  arc(262, 81, 180, 360, 30);
  line(232, 81, 232, 111);
  line(232, 111, 292, 111);
  line(292, 111, 292, 81);
  putpixel(262, 81, 1);
  floodfill(262, 110, 3);

```

```

moveto(262,124);outtext('3');

outtextXY(150,152,'Pilih No =');
GotoXY(31,20);
readln(pilih);
case pilih of
#49: begin
    ClearDevice;
    SetTextStyle(Smallfont,horizDir,3);
    OuttextXY(280,20,'Kang_IS');
    Delay(1000);
    SetTextStyle(DefaultFont,HorizDir,0);
    rectangle(3,20,317,170);
    rectangle(1,18,319,172);
    Line(3,133,317,133);
    arc(60,80,0,180,50);
    line(10,80,10,120);
    line(10,120,110,120);
    line(110,120,110,80);
    line(30,48,30,120);
    line(80,42,80,120);
    putpixel(60,80,1);
    floodfill(60,83,3);
    outtextxy(12,95,'Y');
    outtextxy(12,122,'X1');
    outtextxy(100,122,'X2');
    outtextxy(160,40,'Besar(mm) :');
    outtextxy(170,64,'-. Y =');
    gotoxy(31,9);
    readln(Y);
    outtextxy(170,80,'-. X1 =');
    gotoxy(31,11);
    readln(x1);
    outtextxy(170,96,'-. X2 =');
    gotoxy(31,13);
    readln(x2);
    outtextxy(170,112,'-. R =');
    gotoxy(31,15);
    readln(R);
    akhir := 2*R - X2;
    for i:=Round(x1) to Round(akhir) do
    begin
        yreal := sqr(R) - sqr(R-i);
        yreal1 := round(sqrt(yreal));
        sumbuY := Y + Yreal1;
        DataBuffer[counter] := sumbuY;
        inc(counter);
    end;
    outtextxy(150,152,'Lagi (y/t) ?');
    gotoxy(33,20);
    readln(lagi);
    case lagi of
    'y': MenuUtama;
    't': tampilData;
    end;
end;

```

```

end;

#50: begin
  ClearDevice;
  SetTextStyle(SmallFont, HorizDir, 3);
  OutTextXY(280, 20, 'Kang_1S');
  delay(1000);
  SetTextStyle(DefaultFont, HorizDir, 0);
  Rectangle(3, 20, 317, 170);
  rectangle(1, 18, 319, 172);
  Line(3, 133, 317, 133);
  line(20, 30, 20, 120);
  line(20, 120, 120, 120);
  line(120, 120, 120, 80);
  line(120, 80, 20, 30);
  floodfill(60, 90, 3);
  outtextxy(5, 75, 'Y1');
  outtextxy(123, 100, 'Y2');
  outtextxy(80, 125, 'X');
  outtextxy(160, 40, 'Besar(mm) :');
  outtextxy(170, 64, '-. Y1 =');
  gotoxy(31, 9);
  readln(Y1);
  outtextxy(170, 80, '-. Y2 =');
  gotoxy(31, 11);
  readln(Y2);
  outtextxy(170, 96, '-. X =');
  gotoxy(31, 13);
  readln(X);
  kurang := Y1 - Y2;
  if Y1 > Y2 then
    begin
      sudut := (Y1 - Y2) / X;
      teta := arctan(sudut);
      for i := 0 to Round(X) do
        begin
          bulat := round(sin(teta) / cos(teta));
          sumbuY := Y1 - i * sudut;
          dataBuffer[counter] := sumbuY;
          inc(counter);
        end;
      outtextXY(150, 152, 'lagi (y/t) ?');
      gotoxy(33, 20);
      readln(lagi);
      case lagi of
        'y' : MenuUtama;
        't' : tampilData;
      end;
    end
  else
    begin
      sudut := (Y2 - Y1) / X;
      teta := arctan(sudut);
      for i := 0 to round(X) do
        begin

```

```

        bulat := round(sin(teta)/cos(teta));
        sumbuY := i*sudut + Y1;
        dataBuffer[counter] := sumbuY;
        inc(counter);
    end;
    outtextxy(150,152,'Lagi (y/t) ?');
    gotoxy(33,20);
    readln(lagi);
    case lagi of
        'y':MenuUtama;
        't':tampilData;
    end;
end
end;
end;

#51: begin
    ClearDevice;
    SetTextStyle(SmallFont,HorizDir,3);
    OutTextXY(280,20,'Kang_1S');
    Delay(1000);
    SetTextStyle(DefaultFont,HorizDir,0);
    Rectangle(3,20,317,170);
    rectangle(1,18,319,172);
    Line(3,133,317,133);
    arc(60,30,180,360,50);
    line(10,30,10,120);
    line(10,120,110,120);
    line(110,120,110,30);
    line(30,62,30,120);
    line(80,67,80,120);
    putpixel(60,30,1);
    floodfill(60,83,3);
    outtextxy(12,90,'Y');
    outtextxy(12,125,'X1');
    outtextxy(100,125,'X2');
    outtextxy(160,40,'Besar(mm) :');
    outtextxy(170,64,'-. Y =');
    gotoxy(31,9);
    readln(Y);
    outtextxy(170,80,'-. X1 =');
    gotoxy(31,11);
    readln(x1);
    outtextxy(170,96,'-. X2 =');
    gotoxy(31,13);
    readln(x2);
    outtextxy(170,112,'-. R =');
    gotoxy(31,15);
    readln(r);
    akhir := 2*R - X2;
    for i := Round(X1) to Round(akhir) do
    begin
        yreal := sqr(R) - sqr(R-i);
        yreal1 := round(sqrt(yreal));
        sumbuY := Y - Yreal1;
        DataBuffer[counter] := sumbuY;
    end;
end;

```

```

        inc(counter);
    end;
    outtextxy(150,152,'Lagi (y/t) ?');
    gotoxy(33,20);
    readln(lagi);
    case lagi of
        'y': MenuUtama;
        't': tampilData;
    end;
end;
end;
end;
end;

```

```

{*****}
{          PROCEDURE PENGISIAN DATA KE MEMORY          }
{*****}

```

```

Procedure DataMemori;
var
    maks,
    k          : byte;

```

```

Begin
    maks:=(maks mod 256);
    For k := 1 to maks do
        Begin

```

```

{*****}
{          pengisian data model          }
{*****}

```

```

        DataRam := 50 - trunc(DataBuffer[k]);
        Port[Alamat] := k;
        Port[DtaRam] := DataRam;
    End;
    for k := (maks+1) to $7f do
        begin
            port[alamat] := k;
            port[DtaRam] := $ff;
        end;

```

```

{*****}
{          pengisian panjang maksimum          }
{*****}

```

```

        Port[Alamat] := $7E;
        Port[DtaRam] := maks;

```

```

{*****}
{          pengisian diameter minimum          }
{*****}

```

```

        k := 1;
        Mini := DataBuffer[k];
        Repeat

```



```

If mini < DataBuffer[k+1] then
Mini := DataBuffer[k];
Mini := DataBuffer[k+1];
K := K + 1;
until k = maks;
Port[alamat] := $7f;
port[Dtaram] := trunc(mini);
SetViewPort(4,21,316,169,ClipOn);
ClearViewPort;
SetViewPort(0,0,GetMaxX,GetMaxY,ClipOn);
for k := 1 to $7f do
begin
    port[alamat] := k;
    ata := port[dtaram];
    Str(Ata:4,AtaString);
    OutTextXY(5*TextWidth('S')*(k mod 8),
    TextHeight('H')*(k div 8)+25,ataString);
end;
SetTextStyle(defaultfont,HorizDir,1);
SetUserCharSize(1,2,1,2);
OutTextXY((GetMaxX div 2)-
(TextWidth('Proses Selesai') div 2),
GetMaxY-(TextHeight('H')+5),'PROSES SELESAI');
Repeat
    Pilihh :=ReadKey;
Until Pilihh = Char(27);
End;

```

```

{*****}
{          PROCEDURE TAMPIL GAMBAR          }
{*****}
procedure TampilGambar;
begin
    SetGraphMode(Mode);
    SetTextStyle(SmallFont,HorizDir,3);
    OutTextXY(280,20,'Kang_1S');
    Delay(1000);
    Rectangle(3,20,317,170);
    Rectangle(1,18,319,172);
    Line(3,133,317,133);
    OuttextXY(22,52,'(mm)');
    line(30,60,30,110);
    Line(28,60,30,60);
    OuttextXY(12,55,'50');
    line(30,110,280,110);
    OuttextXY(270,100,'(mm)');
    Line(80,110,80,112);
    OuttextXY(75,117,'50');
    Line(130,110,130,112);
    OuttextXY(125,117,'100');
    Line(180,110,180,112);
    OuttextXY(175,117,'150');
    Line(230,110,230,112);
    OuttextXY(225,117,'200');

```

```

Line(280,110,280,112);
OuttextXY(275,117,'250');
maksi := counter - 1;
MoveTo(30,115);
for k := 0 to maksi do
begin
    sumbuX := 30 + k;
    sumbuY := 110 - dataBuffer[k];
    LineTo(Round(sumbuX),Round(sumbuY));
end;
LineTo(Round(sumbuX),110);
floodfill(32,108,3);
SetTextStyle(DefaultFont,HorizDir,0);
OutTextXY(25,144,'Model sudah benar (y/t) ?');
GotoXY(30,19);
Readln(model);
Case model of
'y' : {Pembubutan;}
    Begin
        OutTextXY(11,160,'Simpan Data Model di Memori (y/t) ?');
        GotoXY(38,21);
        Readln(Simpan);
        if UpCase(Simpan)='Y' then DataMemori;
        End;
't' : begin
        clearDevice;
        counter :=1;
        MenuUtama;
        readln;
        TampilGambar;
        end;
end;
end;

```

```

{*****}
{          PROGRAM UTAMA          }
{*****}
begin
    clrscr;
    driv := cga;
    mode := cgaci;
    initgraph(driv,mode,'');
    SetGraphMode(mode);
    SetTextStyle(smallFont,HorizDir,3);
    OutTextXY(280,20,'Kang_IS');
    Delay(1000);
    SetTextStyle(defaultFont,HorizDir,2);
    Rectangle(3,20,317,170);
    Rectangle(1,18,319,172);
    Line(3,133,317,133);
    OutTextXY(100,40,'PROGRAM');
    OutTextXY(20,65,'PENGISI DATA MODEL');
    OutTextXY(110,90,'di RAM');
    SetTextStyle(DefaultFont,HorizDir,0);

```

```
OutTextXY(100,152,'Tunggu .....');  
delay(5000);  
counter := 1;  
MenuUtama;  
tampilGambar;  
end.
```

\*\*\*\*\*

# PROGRAM KONTROL MESIN BUBUT

Oleh:

ISRAMSYAH

2852200383

\*\*\*\*\*

## Keterangan:

P27 - on(1) off(0) motor x  
P26 - maju(1) mundur(0) motor x  
P25 - on(1) off(0) motor y  
P24 - turun(1) naik(0) motor y  
P23 - on(1) off(0) motor utama

P10 - start switch/ ketebalan bahan  
- switch posisi y naik

P11 - optocoup home posisi y  
- switch posisi y turun

P12 - optocoup home posisi x  
- switch posisi x maju

P13 - stop switch ketebalan bahan  
- switch selesai

P14 - switch feed A  
- switch otomatis

P15 - switch feed B  
- switch manual

P16 - switch feed C  
- switch posisi x mundur

P17 - optocoup posisi x

Alamat 7dh dari RAM berisi jumlah data model.

Alamat 7eh dari RAM berisi diameter data.  
minimum.

Alamat 00h dari RAM berisi data feeding.

Alamat 80h dari RAM berisi data ketebalan  
bahan.

=====

## PROGRAM UTAMA

=====

0000                   ORG    0h  
0000 E5               SEL    MB0  
0001 0410             JMP    KEMBALI  
0003                   ORG    03h

; Rutin interrupt dilayani disini.  
; Saat interrupt, kontroller akan menghentikan  
; mesin, sampai switch start ditekan lagi.

0003                   INTRP:  
0003 D5               SEL    RB1  
0004 A8               mov r0, a  
0005 2308             mov a, #08h  
0007 3A               OUTL   P2, A  
0008                   BACK:  
0008 09               IN     A, P1  
0009 5301             ANL    A, #01h  
000B 9608             JNZ    BACK

```

000D F8      MOV    A, R0
000E C5      SEL    RBO
000F 93      RETR

;
; Program utama dimulai disini
;
0010      KEMBALI:
0064      ORG     64h
0064 05      EN     1
0065 2300    MOV    A, #00h
0067 3A      OUTL   P2, A
0068 14B8    CALL   MULA1
006A      OTOMATIS:
006A 09      IN     A, P1      ; switch manual ditekan
006B 5320    ANL    A, #20h    ; tidak, ke otomatis1
006D 9671    JNZ    OTOMATIS1 ; ya ke rutin manual
006F 3492    CALL   MANUAL
0071      OTOMATIS1:
0071 09      IN     A, P1      ; switch otomatis ditekan
0072 5310    ANL    A, #10h    ; tidak, ke otomatis
0074 966A    JNZ    OTOMATIS  ; ya, operasi otomatis
0076 14BE    CALL   HOMEY
0078 14CC    CALL   HOMEX
007A 14DA    CALL   TEBAL      ; dilakukan
007C      FEED1:
007C 09      IN     A, P1
007D 5310    ANL    A, #10h
007F 9685    JNZ    FEED2
0081 BF02    MOV    R7, #02h
0083 0495    JMP     FEED
0085      FEED2:
0085 09      IN     A, P1
0086 5320    ANL    A, #20h
0088 968E    JNZ    FEED3
008A BF03    MOV    R7, #03h
008C 0495    JMP     FEED
008E      FEED3:
008E 09      IN     A, P1
008F 5340    ANL    A, #40h
0091 967C    JNZ    FEED1
0093 BF04    MOV    R7, #04h
0095      FEED:
0095 09      IN     A, P1
0096 5301    ANL    A, #01h
0098 9695    JNZ    FEED
009A B800    MOV    R0, #00h    ; simpan data feeding
009C FF      MOV    A, R7      ; ke RAM
009D 90      MOVX   @R0, A
009E      KEMBALI3:
009E 37      CPL     A
009F 0301    ADD     A, #01h
00A1 AC      MOV     R4, A
00A2 04F8    JMP     MAKAN
00A4      HOOP5:
00A4 B880    MOV     R0, #80h

```

```

00A6 80      MOVX  A, @R0
00A7 AA      MOV   R2, A
00A8 B87E    MOV   R0, #7Eh
00AA 80      MOVX  A, @R0
00AB 6C      ADD   A, R4
00AC 6A      ADD   A, R2
00AD F6B1    JC    HOOP51
00AF 24ED    JMP    KELUAR
00B1          HOOP51:
00B1 B800    MOV   R0, #00h
00B3 80      MOVX  A, @R0
00B4 6F      ADD   A, R7
00B5 AF      MOV   R7, A
00B6 049E    JMP    KEMBALI3

;
00B8          MULAI:
00B8 09      IN     A, P1      ; input start switch
00B9 5301    ANL    A, #01h
00BB 96B8    JNZ    MULAI
00BD 83      RET

;
=====
SUBROUTIN HOME POSISI Y
=====
;
00BE          HOMEY:
00BE 09      IN     A, P1      ; input optocoup homeY
00BF 5302    ANL    A, #02h    ; aktif ke stop1
00C1 C6C8    JZ     STOP1
00C3 2320    MOV    A, #20h    ; tidak, motorY naik
00C5 3A      OUTL   P2, A
00C6 04BE    JMP    HOMEY
00C8          STOP1:
00C8 2300    MOV    A, #00h
00CA 3A      OUTL   P2, A
00CB 83      RET

;
=====
SUBROUTIN HOME POSISI X
=====
;
00CC          HOMEX:
00CC 09      IN     A, P1      ; input optocoup homeX
00CD 5304    ANL    A, #04h    ; aktif, ke stop2
00CF C6D6    JZ     STOP2
00D1 2380    MOV    A, #80h    ; tidak, motorX mundur
00D3 3A      OUTL   P2, A
00D4 04CC    JMP    HOMEX
00D6          STOP2:
00D6 2300    MOV    A, #00h
00D8 3A      OUTL   P2, A
00D9 83      RET

;
=====
SUBROUTIN KETEBALAN BAHAN
=====
;

```

```

00DA      TEBAL:
00DA 09      IN    A, P1      ; input switch ketebalan
00DB 5301    ANL    A, #01h   ; bahan
00DD C6E1    JZ     HOOP88
00DF 04DA    JMP    TEBAL
00E1      HOOP88:
;
00E1 2300    MOV    A, #00h   ; Acc = 000h
00E3 62      MOV    T, A
00E4 45      STRT   CNT       ; start counter
;
00E5 2330    MOV    A, #30h   ; motor Y turun
00E7 3A      OUTL   P2, A
00E8      CEK1:
00E8 09      IN     A, P1     ; input switch ditekan
00E9 5308    ANL    A, #08h   ; lagi
00EB 96E8    JNZ    CEK1
;
00ED 2300    MOV    A, #00h   ; motor Y berhenti
00EF 3A      OUTL   P2, A
;
; Second complement data counter
00F0 42      MOV    A, T      ; pindahkan isi counter
00F1 37      CPL    A        ; ke acc.
00F2 0301    ADD    A, #01h
00F4 B880    MOV    R0, #80h  ; pindahkan ke RAM
00F6 90      MOVX   @R0, A
;
00F7 83      RET
;
; =====
; SUBROUTIN PEMAKANAN
; =====
MAKAN:
00F8 B87D    MOV    R0, #7dh   ; ambil jumlah data maks
00FA 80      MOVX   A, @R0
00FB AB      MOV    R3, A      ; pindahkan ke regs.3
00FC B880    MOV    R0, #80h
00FE 80      MOVX   A, @R0
00FF AA      MOV    R2, A
0100 B801    MOV    R0, #01h
0102 B981    MOV    R1, #81h
0104 00      NOP
0105 00      NOP
0106 00      NOP
0107 00      NOP
0108      LOOP4:
0108 80      MOVX   A, @R0     ; ambil data model
0109 6A      ADD    A, R2     ; kurangkan dg tebal bahan
010A 6C      ADD    A, R4     ; kurangkan dg feeding
;
010B F613    JC     BESDATA   ; data lebih besar dari
010D 80      MOVX   A, @R0     ; feeding, ambil data
010E 6A      ADD    A, R2     ; model, kurangkan dg
010F 37      CPL     A        ; tebal bahan

```

0110 17		INC	A	
0111 2414		JMP	DATAIN	; tidak data tetap
0113	BESDATA:			
0113 FC		MOV	A, R4	
0114	DATAIN:			
0114 91		MOVX	@R1, A	
0115 18		INC	R0	
0116 19		INC	R1	
0117 CB		DEC	R3	
0118 FB		MOV	A, R3	
0119 C61D		JZ	HOOP31	
011B 2408		JMP	LOOP4	
011D	HOOP31:			
				; motor Y maju sebesar feeding
011D 2300		MOV	A, #00h	
011F 62		MOV	T, A	
0120 45		STRT	CNT	
0121 2330		MOV	A, #30h	
0123 3A		OUTL	P2, A	; motor Y turun
0124	HITUNG1:			
0124 42		MOV	A, T	; input counter
0125 6C		ADD	A, R4	; bandingkan dg
0126 9624		JNZ	HITUNG1	; feeding
0128 2300		MOV	A, #00h	
012A 3A		OUTL	P2, A	; motor Y berhenti
012B B87D		MOV	R0, #7dh	
012D 80		MOVX	A, @R0	
012E AB		MOV	R3, A	
012F B981		MOV	R1, #81h	
0131	ULANG2:			
0131 81		MOVX	A, @R1	
0132 AE		MOV	R6, A	; simpan data di R6
0133	CEK3:			
0133 42		MOV	A, T	; ambil data counter
0134 6E		ADD	A, R6	; bandingkan dg data RAM
0135 9639		JNZ	HOOP44	
0137 2444		JMP	HOOP	
0139	HOOP44:			
0139 E63D		JNC	HOOOP	
013B 2446		JMP	MY_NAIK	
013D	HOOOP:			
013D 2330		MOV	A, #30h	
013F 3A		OUTL	P2, A	
0140	BALIK52:			
0140 42		MOV	A, T	
0141 6E		ADD	A, R6	
0142 9640		JNZ	BALIK52	
0144	HOOP:			
0144 245D		JMP	MY_STOP	; sama motorX berhenti
0146	MY_NAIK:			
0146 2320		MOV	A, #20h	
0148 3A		OUTL	P2, A	; motorY naik
0149	LOOP12:			



0149 42		MOV	A, T	; decrement counter
014A AD		MOV	R5, A	
014B	LOOP6:			
014B 42		MOV	A, T	
014C 37		CPL	A	
014D 0301		ADD	A, #01h	
014F 6D		ADD	A, R5	
0150 9654		JNZ	HOOP6	
0152 244B		JMP	LOOP6	
0154	HOOP6:			
0154 42		MOV	A, T	
0155 03FE		ADD	A, #0FEh	
0157 62		MOV	T, A	
0158 6E		ADD	A, R6	; bandingkan count. dg RAM
0159 C65D		JZ	MY_STOP	
015B 2449		JMP	LOOP12	
015D	MY_STOP:			
015D 23C0		MOV	A, #0C0h	; motor Y berhenti
015F 3A		OUTL	P2, A	; motor X maju
0160 00		NOP		
0161 00		NOP		
0162 09		IN	A, P1	; input optocoup posisi X
0163 5380		ANL	A, #80h	; aktif?
0165 965D		JNZ	MY_STOP	
0167	BALIK22:			
0167 09		IN	A, P1	
0168 5380		ANL	A, #80h	
016A C667		JZ	BALIK22	
016C 2300		MOV	A, #00h	
016E 3A		OUTL	P2, A	
	;			
016F 19		INC	R1	
0170 CB		DEC	R3	
0171 FB		MOV	A, R3	
0172 C676		JZ	HOOP2	
0174 2431		JMP	ULANG2	
0176	HOOP2:			
0176 2320		MOV	A, #20h	; motor Y naik
0178 3A		OUTL	P2, A	
0179	LOOP11:			
0179 42		MOV	A, T	; ambil counter
017A AD		MOV	R5, A	
017B	LOOP7:			
017B 42		MOV	A, T	
017C 37		CPL	A	
017D 17		INC	A	
017E 6D		ADD	A, R5	
017F C67B		JZ	LOOP7	
0181 42		MOV	A, T	
0182 03FE		ADD	A, #0FEh	; decrement
0184 62		MOV	T, A	
0185 53FF		ANL	A, #0FFh	
0187 C68B		JZ	LOOP8	
0189 2479		JMP	LOOP11	
018B	LOOP8:			

```

018B 2300          MOV    A, #00h      ; sama motor berhenti
018D 3A            OUTL   P2, A
018E 14CC          CALL   HOMEX        ; kembali ke X awal
0190 04A4          JMP    HOOP5

;
; operasi manual dilakukan disini
;
0192          MANUAL:
0192 09            IN     A, P1        ; switch Y turun ditekan
0193 5302          ANL    A, #02h      ; tidak, kmanual1
0195 96A4          JNZ    MANUAL1
0197 2330          MOV    A, #30h      ; motor Y turun
0199 3A            OUTL   P2, A
019A 34E3          CALL   DELAY
019C          BALIK:
019C 09            IN     A, P1
019D 5302          ANL    A, #02h
019F C69C          JZ     BALIK
01A1 2300          MOV    A, #00h
01A3 3A            OUTL   P2, A
01A4          MANUAL1:
01A4 09            IN     A, P1        ; switch X maju ditekan
01A5 5304          ANL    A, #04h      ; tidak, kmanual2
01A7 96B6          JNZ    MANUAL2
01A9 23C0          MOV    A, #0C0h     ; motor x maju
01AB 3A            OUTL   P2, A
01AC 34E3          CALL   DELAY
01AE          BALIK1:
01AE 09            IN     A, P1
01AF 5304          ANL    A, #04h
01B1 C6AE          JZ     BALIK1
01B3 2300          MOV    A, #00h
01B5 3A            OUTL   P2, A
01B6          MANUAL2:
01B6 09            IN     A, P1        ; switch Y naik ditekan
01B7 5301          ANL    A, #01h      ; tidak, ke manual3
01B9 96C8          JNZ    MANUAL3
01BB 2320          MOV    A, #20h      ; motor y naik
01BD 3A            OUTL   P2, A
01BE 34E3          CALL   DELAY
01C0          BALIK2:
01C0 09            IN     A, P1
01C1 5301          ANL    A, #01h
01C3 C6C0          JZ     BALIK2
01C5 2300          MOV    A, #00h
01C7 3A            OUTL   P2, A
01C8          MANUAL3:
01C8 09            IN     A, P1        ; switch X mundur ditekan
01C9 5340          ANL    A, #40h      ; tidak, ke manual4
01CB 96DA          JNZ    MANUAL4
01CD 2380          MOV    A, #80h      ; motor X mundur
01CF 3A            OUTL   P2, A
01D0 34E3          CALL   DELAY
01D2          BALIK3:
01D2 09            IN     A, P1

```

```

01D3 5340          ANL    A, #40h
01D5 C6D2          JZ     BALIK3
01D7 2300          MOV    A, #00h
01D9 3A            OUTL   P2, A
01DA              MANUAL4:
01DA 09            IN     A, P1      ; switch selesai ditekan
01DB 5308          ANL    A, #08h    ; tidak, ke manual
01DD C6E1          JZ     HOOP99
01DF 2492          JMP    MANUAL
01E1              HOOP99:
01E1 24ED          JMP    KELUAR      ; lompat Kesselesai
;
;=====
;                      SUBROUTIN DELAY
;=====
; Subrutin delay digunakan untuk menghilangkan
; efek debounce pada saat penekanan switch.
01E3              DELAY:
01E3 27            CLR    A
01E4 62            MOV    T, A
01E5 55            STRT   T
01E6              LOOP0:
01E6 42            MOV    A, T
01E7 B2EB          JB5    TERUS
01E9 24E6          JMP    LOOP0
01EB              TERUS:
01EB 65            STOP   TCNT
01EC 83            RET
;
01ED              KELUAR:
01ED 2308          MOV    A, #08h
01EF 3A            OUTL   P2, A
01F0              ULANG45:
01F0 09            IN     A, P1
01F1 5308          ANL    A, #08
01F3 96F0          JNZ    ULANG45
0000              END

```

\*\*\*\*\* S Y M B O L   R E F E R E N C E   T A B L E \*\*\*\*\*

0008 = BACK	019C = BALIK	01AE = BALIK1
01C0 = BALIK2	0167 = BALIK22	01D2 = BALIK3
0140 = BALIK52	0113 = BESDATA	00B8 = CEK1
0130 = CEK3	0114 = DATAIN	01E3 = DELAY
0095 = FEED	007C = FEED1	00B5 = FEED2
008E = FEED3	0124 = HITUNG1	00C0 = HOMEX
00BE = HOMEY	013D = HOOP	0144 = HOOP
0176 = HOOP2	011D = HOOP31	0139 = HOOP44
00A4 = HOOP5	00B1 = HOOP51	0154 = HOOP6
00E1 = HOOP88	01E1 = HOOP99	00C3 = INTRP
01ED = KELUAR	0010 = KEMBALI	009E = KEMBALI3

01E6 = LOOP0  
0108 = LOOP4  
018B = LOOP8  
01A4 = MANUAL1  
01DA = MANUAL4  
015D = MY\_STOP  
00C8 = STOP1  
01EB = TERUS

0179 = LOOP11  
014B = LOOP6  
00F8 = MAKAN  
01B6 = MANUAL2  
00B8 = MULAI  
006A = OTOMATIS  
00D6 = STOP2  
0131 = ULANG2

0149 = LOOP12  
017B = LOOP7  
0192 = MANUAL  
01C8 = MANUAL3  
0146 = MY\_NAIK  
0071 = OTOMATIS1  
00DA = TEBAL  
01F0 = ULANG45

TOTAL SYMBOLS DEFINED = 57

NO ERRORS FOUND  
END OF ASSEMBLY

01-01-1980  
00:52:59

## USULAN TUGAS AKHIR

1. JUDUL TUGAS AKHIR : PERENCANAAN DAN PEMBUATAN ALAT  
KONTROL KERJA MESIN BUBUT  
DENGAN MENGGUNAKAN  
MIKROKOMPUTER M8035L
2. BIDANG STUDI : Elektronika
3. RUANG LINGKAP :
  - Elektronika analog
  - Elektronika digital
  - Elektronika industri
  - Elektronika mikro
4. LATAR BELAKANG : Pembuatan benda kerja dengan menggunakan mesin bubut dengan cara yang konvensional akan membutuhkan waktu yang relatif lama dan memerlukan kejelian mata maupun ketelatenan untuk mendapatkan hasil kerja yang presisi, meskipun hal itu sulit ditemui.  
  
Dengan adanya kontroller menggunakan mikrokomputer maka diharapkan bisa mendapatkan hasil kerja yang sesuai dengan yang diharapkan dalam waktu yang relatif lebih singkat.

5. PENELAAHAN STUDI : Untuk membuat model dari benda kerja yang akan dibubut maka digunakan komputer IBM PC. Kemudian data tersebut dipindahkan ke RAM dan data tersebut akan diolah oleh bagian kontroller untuk dibandingkan dengan posisi tool dan jika posisi tersebut tidak sama maka kontroler akan menggerakkan tool sampai posisi tool sama dengan data di RAM. Untuk menggerakkan tool baik pada posisi pemakanan maupun pengaturan digunakan motor dc yang akan bergerak dalam dua arah. Sedangkan untuk mengetahui posisi dari tool digunakan optocoupler.

6. TUJUAN : Untuk mendapatkan hasil kerja yang lebih presisi dalam waktu yang relatif lebih singkat tanpa merusak bahan.

7. LANGKAH-LANGKAH : Studi literatur  
Perencanaan peralatan  
Pembuatan peralatan

Pengujian dan kalibrasi

Penyusunan naskah

8. JADWAL KEGIATAN : Seluruh kegiatan direncanakan dapat diselesaikan dalam waktu enam bulan dengan jadwal sebagai berikut.

LANGKAH KEGIATAN	BULAN					
	I	II	III	IV	V	VI
STUDI LITERATUR						
PERENC. ALAT						
PEMBUATAN ALAT						
UJI DAN KAL.						
SUSUN NASKAH						

9. RELEVANSI : Diharapkan dengan alat kontrol ini memperingan pekerjaan dari operator mesin bubut dalam menyelesaikan tugasnya.